EDLnet

D2.5

Europeana Outline Functional Specification

For development of an operational European Digital Library

Authors:

Makx Dekkers, Stefan Gradmann, Carlo Meghini

Contributors and peer reviewers:

EDLnet WP2 Working Group members, EDLnet office

Date:

Version: 1.0

Table of Contents

1 Introduction

The present document is one of the main results of EDLnet[1] WP2. It contains the functional and to some extent also technical) specifications of Europeana, the European Digital Library.

2 Functional Specification Principles & Methodology Background and context

2.1 Methodology

2.1.1 Use Case Analysis

2.1.2 Use Case Management

2.1.3 Use-Case Model layout

3 User Groups and Requirements[2]

3.1 End users[3]

3.1.1 Group Characteristics

Workpackage 3 has identified five user profiles for end users of the Europeana service:

- General User
- School Child
- Academic user (both students and teachers)
- Expert Researcher
- Professional user, e.g. librarian, archivist, etc

These user groups can be characterised as follows:

The ***general user*** has a generic interest in culture or history. He is familiar with basic search functionalities, has no specific domain knowledge, is 'google-minded' and visits sites that have large volumes of content to offer, such as YouTube[4] and Wikipedia[5].

The ***school child*** will make use of the service as part of educational exercises. Culture and heritage are incorporated in many school curricula, which means that Europeana

---

1 EDLnet project: http://www.europeandigitallibrary.eu/edlnet/

2 For the time being se cases are defined for end users only!

3 This section is based on information derived from the following documents: Report on the WP1 meeting of 17-18 December 2007; Draft deliverable D3.1 User Use Cases: Functional requirements for EDL Maquette, dated 6 February 2008; The DRIVER Functional Specification, dated 15 September 2006; "Klik naar het verleden" (see: http://www.scp.nl/publicaties/boeken/9037702791/Klik_naar_het_verleden.pdf)

4 YouTube - http://www.youtube.com/

5 Wikipedia: http://www.wikipedia.org/

could be used in a variety of educational contexts. The school child will expect the service to be easily accessible, immediately appealing, visually attractive or even playful, no jargon and easy to handle while dealing with their exercises.

The ***academic user*** represents the other end of the educational spectrum. He may have excellent domain knowledge, or aspires to achieve that. He will expect the information offered to be comprehensive, accurate, representative if not complete, and easy to reuse in the context of educational assignments.

The ***expert researcher*** looks for specific information on a specific topic. He is to a certain degree skilled in using retrieval services and may make use of the advanced search button to get the most out of the system. As this group is most likely to publish the results of the research in one way of the other, this group includes users who are prepared to buy something or travel to visit the contributing institutions.

The ***professional user*** is most likely a staff member of a cultural heritage organisation. He is skilled in using information systems, but with a different perspective than expert researchers. He may be interested in details as well as very generic information, for instance for improving information services of his own institution.

If we look at the motives of these user groups to use Europeana, it is possible to identify four types of objectives:[1]

1. The user wants to be entertained

This includes users who have time on their hands to browse around the Internet and have a structural or incidental interest in cultural heritage. They come to Europeana because they expect that there is a lot of interesting content. For these users it is not important what they find is as long as it is interesting and entertaining.

2. The user wants to know more about a cultural or historic subject or person

This includes users that have a specific reason for their interest: it could be that they need to do a project for school, study or work on a certain subject, or that they have been made aware of a certain subject through current news or in a conversation with colleagues, friends or family. These users are looking for the most relevant results and would not want to see lots of results that are not relevant to them. To be able to determine what is relevant to them, information about the specific objective of the user is necessary.

3. The user wants to know the current whereabout of cultural heritage

This includes users that are planning to see the original objects for research purposes, or users that are about to undertake a trip and would like to know what cultural heritage they can visit during a touristic trip or other type of stay. These persons will also be interested in getting more information on interesting events and collections in the area, asl well as local services such as guided tours.

4. The user wants to be part of a community of interest

This includes users who may be students, researchers or members of a cultural society and want to share their knowledge via an online environment such as a social platform with a cultural focus. They may want to present their opinions and ratings of cultural heritage resources to their kin as well as share personal items (photographs, documents etc.)

3.1.2 Group Expectations

As a large scale international information service on culture, history and heritage, Europeana will deal with a large variety of user expectations: from one click amusement to in-depth research facilities.

One thing all end-users have in common is that they want access to the Europeana full content through *search and browse* and *direct surrogate addressing and access* options. Which option (or configuration of that option) they will choose depends on the user profile (general user, school child, researcher) and objective (entertainment, research, community building).

End-users also expect to *interact* with the content: *view* and additionally *download* a film footage; *copy and paste* information for a paper they are writing; *create sets* of preferred items in the Europeana collection; *study details* of high resolution reproductions of cultural objects; *upload* a personal item; *contribute* to enrich the description of materials through social tagging; *share* interests and comments about the cultural heritage items within communities.

Given the expectations generally described above, the search and retrieval requirements for the Europeana services can be detailed as follows:

- Search
    - to do a free text search from a single text field form
    - to get help about the semantic net of the chosen descriptor by e. g. a topic map feature
    - to do an free text search with a variety of additional search parameters (e.g. language, media type)
    - to use precompiled lists for searching in the four major Europeana facets.
- Results display
    - to get a brief view of search results (both text and thumbnails)
    - to get detailed view for a single search result
    - to get help in the form of search tips and suggestions, especially when there is no result ('did you mean....')
    - to filter search results by specific criteria predefined by the user
    - to get access to full content: read a book, play a video, listen to an audio recording, etc…
    - to get access to similar or related full content
    - to display Europeana results outside the context of the Europeana portal, e.g. in a personal blog or a mash up website
- Browse
    - to browse the content of Europeana by the spatial dimension (where)
    - to browse the content of Europeana by the time dimension (when)

- to browse the content of Europeana by subject, being a "who" or a "what"
- to browse through predefined sets of results on popular subjects

- Personalisation
  - to register for a personal Europeana account
  - to authenticate with Europeana
  - to use and maintain a personal space in Europeana (MyEuropeana), e.g. to store preferred searches, lay out or language.
- Alerting & Saved searches
  - to save search and browse selected results
  - to retrieve stored searches at a later occasion
- Collaborative workspace
  - to start and maintain a community within Europeana
  - to join or leave a Europeana community
  - to forward or share saved search and browse results
  - to link to the profiles of other Europeana user
- User oriented editorial
  - to tag existing content (social tagging)
  - to upload objects from their private collections

3.2 External applications / API

3.2.1 Group Characteristics

This "user group" consists of software programs that may be part of other types of systems or portals that provide aggregated access.

The objective of these applications would in general be to integrate Europeana content in a more or less seamless manner with other resources.

These applications use machine-to-machine communication that will at least need access to the authentication and advanced search functionality of Europeana. In how far they need access to the Europeana surrogates depends on the sophistication of the application.

These external applications will be themselves responsible for accessing the primary resources at content providers.

3.2.2 Group Expectations

The expectations of this group can be:

- to get an HTTP[6] access to the Europeana content (surrogates) and services through a set of specifications described in APIs

[6] HTTP – Hypertext Transfer Protocol: http://www.w3.org/Protocols/

- to request Europeana content according to different criteria: by date, in different formats and schemes, etc…
- [to complete]

3.3 Content aggregators / providers

This user group consists of

a) content providers: organisations which directly provide content to the central Europeana system;

b) content aggregators: organizations that act as collection points for content from other providers, e.g. for smaller institutions or individuals that for one reason or another cannot connect to the central Europeana system to contribute content.

The objective of these users is to contribute content as a proxy to the central Europeana system.

The functions that these users need to access have to do mostly with the collection and provision of compliant data conforming with the Europeana requirements to describe and access content.

3.3.1 Group Expectations

3.3.1.1 Content providers

The interaction of content providers with Europeana lies in the area of data provision (exposure from the repository's side of view) to content aggregators. They must publish the data in appropriate forms to achieve the proper visibility for the repository's content taking into account possible usage restrictions and associated Europeana requirements.

This means submitting content (and associated data) to content aggregators in a controlled and automatic way.

The OAI-PMH protocol[7] is a widely used mechanism employed for this purpose, and Europeana has decided to make this a prerequisite for the data collection procedure and content aggregators will accordingly collect data from content providers. However, this doesn't exclude other technologies (such as P2P) from being considered as additional delivery methods, as long as OAI-PMH is supported as the elementary one.

The content providers may wish to have their branding identity exposed when their content is accessed from within the Europeana portal. But since their content is accessed within their own repositories and they are also providing the applications to expose the content (e.g. page-turning application), they have the means to show their identity along with the digital objects without further assistance from Europeana.

3.3.1.2 Content aggregators

In this context, content aggregators are the interface between the content providers and Europeana.

Their role is

[7] OAI-PMH – Open Archives Initiative Protocol for Metadata Harvesting: http://www.openarchives.org/OAI/openarchivesprotocol.html

- to collect information about providers and their delivery systems.
- to collect data about content being provided as a surrogate
- to de-duplicate, desambiguate, clean, enrich the data with meaningful attributes, possibly associate content in collections
- to verify the accessibility of content
- to make data ready for Europeana data collection using the OAI-PMH protocol

Content aggregator requirements are centred on metadata quality and so services that enhance metadata can be of crucial importance.

3.3.1.3 Europeana expectations

An important functionality Europeana expects from the content providers is to implement within their digital object presentation applications (including simple html pages) mechanisms to expose the Europeana identity when the user accesses the object via the Europeana portal. This could be achieved - for instance - via a script which either:

- "envelopes" the object with a specific Europeana frame or
- "pastes" the Europeana logo on the object

when the user lands on the object from the Europeana portal.

3.3.2 Repository management and data collection functionalities

Europeana has defined the following set of repository management (e.g. registration), as well as to the data collection (e.g. harvesting) functionalities to be supported.

Data Provision

From a content provider perspective, the following data provision/collection functionalities apply when collecting data from content providers and when forwarding data to Europeana:

- Straightforward registration procedure: content providers are provided with client implementations using a simple common metadata registration function, in which they map all necessary information on the characteristics of the repository and the metadata (format, sets and subjects, etc.) from their database. The final approval and actual registration process should be a part of the Europeana system itself.
- Scheduled harvesting: there should be a means of defining an automated process for conducting the data collection at regular intervals;
- Incremental harvesting: in order to minimize traffic and enhance performance in a multi-repository environment, Europeana must take advantage and support incremental harvesting to collect only addtional new available content;
- Refresh harvesting: periodically, even if repositories support incremental publishing methods, there is a need for performing a "fresh start". Europeana must be able to support this function, and should be designed in such a way that this process is fast and causes no complications with the harvesting of other repositories;
- Automatic error handling and reports: in case of errors, the content providers must be notified with detailed error reports. Also they should be able to check the status of the data transfer to the various harvesting points, especially when intermediate aggregators are involved.

Security

- *Access to data repositories should be restricted to authorised content providers*
- *Under certain circumstances, access to content may be restricted although data has been collected signalling its availability. Data collected and forwarded to Europeana should provide sufficient information for the user to contact the content provider. In this case the surrogate could be a link to content related information including content provider contact information or a link to a content access procedure under the content of the content provider.*

Usage

Content providers want to know how much usage is being made of their content and how and where this usage is occurring. This sort of feedback is used by content providers to secure commitment and buy-in from senior management in their institutions. The system must therefore support the following list of features and functionalities:

- Content providers want to control the usage made of content. In some circumstances content may only be consulted (streamed) while in others a personal copy may be permitted. The system should allow to enforce such usage restrictions.
- Harvesting reports and statistics: Europeana must support such functionalities in order to provide system wide quantitative and comparative data back to content providers.
- Metadata usage reports and statistics (since they are able to monitor themselves the actual usage of their own digital objects).
- User profiling: this may provide an indirect mechanism for maintaining the required statistics, and it could be structured in such a way that qualitative measurements are offered;

Support

Data publishers want to be provided with a clear web accessible documentation that states the guidelines and requirements they have to conform to for the delivering of their data: schemes and format, identification/versioning, etc… Also they often wish to be provided with tools for preparing and testing the compatibility of their content and the reliability of their delivery system.

3.4 Service providers

3.4.1 Group Characteristics

This "user group" consists of systems that provide services to Europeana, for example in providing enriched or associated content to be linked to Europeana resources.

The objective of these users is to allow discovery and access to resources that are not described and indexed in the Europeana system.

The functions that interacts with these providers are primarily the Search Gateway (where queries are translated into external searches) and the Results Manager (where content from external sources is integrated with results from within Europeana).

3.4.2 Group Expectations

Service providers need to receive from Europeana the information required for carrying out a specific application task like, e.g., data crawling, searching, harvesting.

They should get access to specifications that explains how to carry on these tasks in the form of APIs and toolkits available online.

3.5 Meta users

3.5.1 Group Characteristics

This user group includes of people who are interested in the performance and visibility of the Europeana service and in statistics related to its content and usage.

These users need access to the system administration functions of Europeana.

3.5.2 Group Expectations

3.5.2.1 Meta users / Administrators

The requirements for this user group are:

- To monitor the Europeana system for:
    - Performance: accessibility, reliability …
    - Traffic for the different user groups and profiles
    - Content: volume of surrogates/original objects/…, number of content providers/aggregators, traceability of data ingestion and distribution
    - Partners: number, domain, country,…
    - Usage of Europeana

  The monitoring frequency and parameters should be configurable.
- To get statistical reports for the monitored data, that can be exported in various formats for re-use and dissemination. Reports can be obtained in both comprehensive and detailed views.
- To access the raw logged data in order to conduct customed analysis. These raw data can be accessed by software application to take into account in configuration settings like, e.g., the optimisation of queries.

Policy makers are a sub-group of administrators: they also need to access the Europeana system to check how it performs, what is the content and who are the contributors but they are mostly interested in evaluation reports with a clear overview of the statistics.

3.6 Policy makers

3.6.1 Group Characteristics

This user group consists of people who are interested in the effects of the Europeana service and its influence on the role that digital heritage plays in social and professional life as a result of the availability of the material through Europeana.

These users will not as a rule interact with the Europeana system directly but need to be provided with status and evaluation reports and in addition, reports describing performance and contributions.

3.6.2 Group Expectations

To be completed.

4 Europeana System Functional Vision

4.1 Europeana Basic Concepts

4.1.1 Actors

Actors that play a role once the Europeana service has become operational, can be identified in four functional areas: access, content provision, system management and service monitoring.

4.1.1.1 Access

End users

These are human actors that access the Europeana services through http://europeana.eu/ or another preferred point of entry that is interoperable with Europeana. They interact with the end user oriented services through the search and browse interfaces and myEuropeana. For some services they need to be identified as registered users, so they can make full use of the services.

External applications

These are systems that connect automatically to the API search interface in order to search or harvest content from Europeana in order to serve users that access external systems. Such external applications may or may not be known and authenticated by Europeana. The same can be said about the users of these services. Depending on the status of service and the user, different services may be provided. To reach flexibility as well as reliability, use of widely supported authentication schemes is required.

4.1.1.2 Content provision

Content providers

These are the organisations that commit to providing resources and associated metadata to be included in the central Europeana indexes and surrogate collections. These are registered organisations whose organisational and technical capabilities are known to Europeana management in some form of service agreement.

Content aggregators

These are a special type of content providers that act as collectors of content from content providers that themselves are not able of willing to contribute to Europeana directly. Towards Europeana, the content aggregators take on the responsibility to ensure quality of resources and metadata through some form of service agreement.

External Service providers

These are organisations that give access to their information systems and services through Europeana. It may be necessary that Europeana concludes service agreements with service providers that provide essential information for Europeana resources.

4.1.1.3 System management

System configuration managers

These are people that secure the configuration of Europeana necessary for realising the diversity of services for different user groups, as well as for the interoperability with external services.

System platform operators

These are people who ensure the day-to-day operation of the Europeana system and services, among other things taking care of backup and restore procedures and scheduling of harvesting, indexing and content loading and summarization.

4.1.1.4 Service monitoring

Customer relations and help-desk

These are people who oversee the service from the perspective of the users of the system and services. They can forward questions related to content to the relevant partner in Europeana. In case of operational problems, they have access to analysis tools and emergency procedures to solve any problems.

Report generators

These are people who ensure that operational statistics and evaluation reports are produced and made available to policy makers and other stakeholders.

4.1.2 Logical data model: Objects and Surrogates

A central principle for building Europeana is that a network of semantic resources will be used as the primary level of user interaction. In a classical librarian catalogue model all user access to information objects is mediated by descriptive metadata as illustrated in Figure 1**Fehler! Verweisquelle konnte nicht gefunden werden.** below:

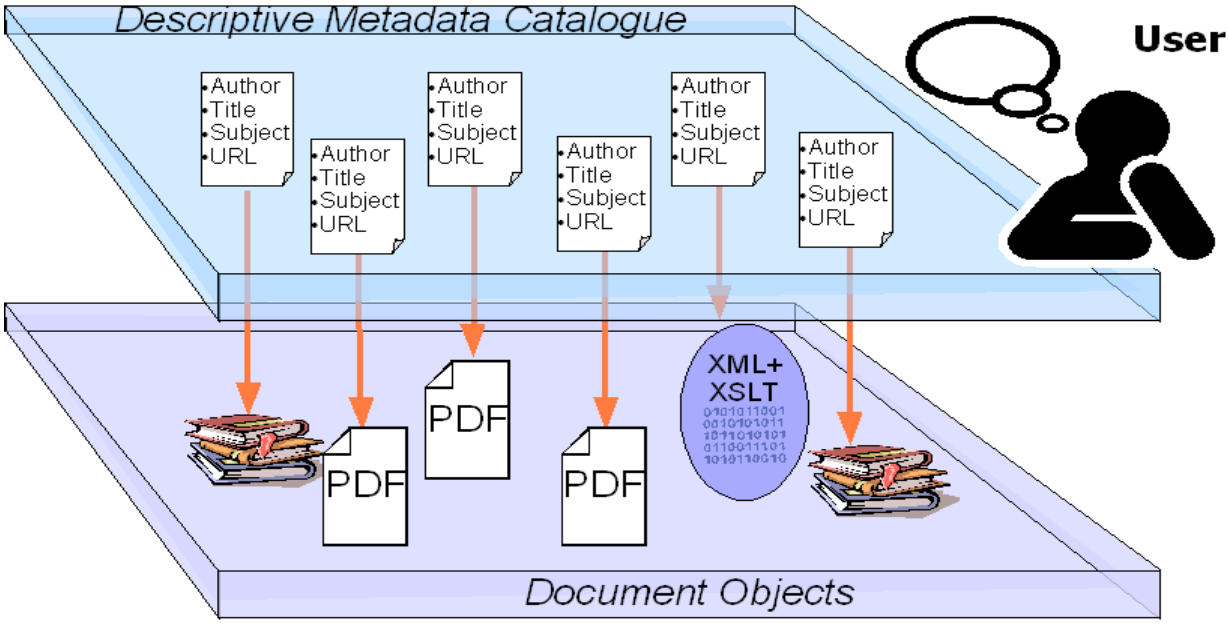

Figure 1: Catalogues and Information Objects in Digital Libraries

Unlike in such librarian functional models users are expected to explore the Europeana data space using semantic nodes as primary elements for searching and browsing along paradigms indicated by the questions as to "Who?", "Where?", "When?" and "What?" The intended relation between the semantic and the object representation layers with respect to the Europeana user interface is illustrated in Figure 2 below:

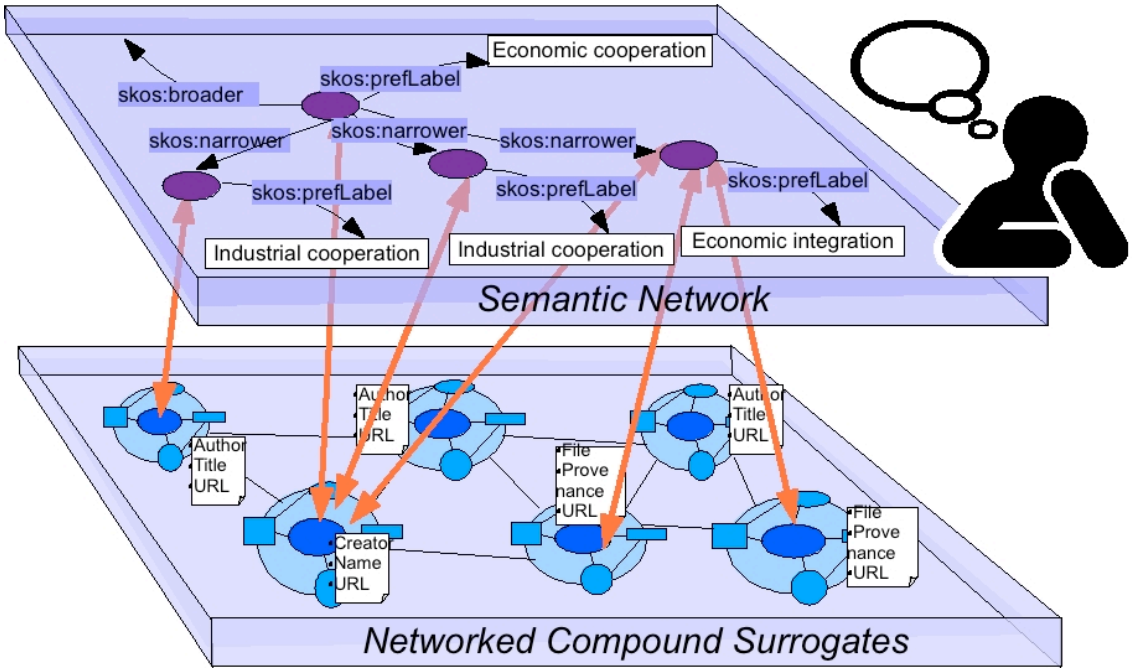

Figure 2: Semantic and Surrogate Layer in Europeana

The user now primarily interacts with the semantic network to explore the Europeana surrogate space which now has the metadata as parts of the surrogates and surrogate aggregations.

In the perspective of this approach, Europeana can be thought of as a ***network of inter-operating object surrogates enabling semantics based object discovery and use***. This network in turn is an integral part of the overall information architecture of the WWW.

Furthermore, the Europeana object model is based on the assumption that the central Europeana data store will only contain object surrogates and index files, whereas original objects are located at the content provider sites. Europeana thus will create a parallel data space inside the system that is a representation of the real world object space. As a consequence, we distinguish 'object entities' (to indicate an external object plus any associated metadata about that object) and 'surrogate entities' (to indicate the internal object with associated metadata and other composite elements). Likewise, two separate data spaces need to be distinguished: an external space of objects entities and an internal space of surrogate entities.

4.1.1.5 Surrogates

All surrogates in the Europeana data space are web resources in the sense defined by the W3C and thus have a URI[8] identifier. They also contain a link to the object entity in case this object can be identified as a web resource. Otherwise the link will be to an external application permitting access to this object. In some specific settings requiring exclusive control by the content provider of all access methods and functionality a surrogate thus can be limited to being an entry to a content access point under control of the content provider.

In such an approach the Europeana surrogate model can be completely agnostic about where the original objects are stored: the URI link to the object syntactically remains

[8] Uniform Resourc Identifier (URI) - http://tools.ietf.org/html/rfc3986

the same. The surrogate model thus isn't affected, in case the option of also keeping original objects within Europeana is needed (for instance for content providers that do not have a content store of their own or for some other reason prefer to store their objects within the Europeana environment): these objects would still be kept in a separate Europeana object store and be referenced from their surrogates just as external objects would be.

The model is conceived from an 'atomic', bottom-up perspective: the basic building blocks are surrogates representing the minimal significant documentary object units a given content provider is able / willing to identify (in the case of textual object there thus can be surrogates on the level of the entire document, on chapter level or on page, paragraph, sentence or even word level).

Each of these surrogates contains at least a URI, a link to the original object, metadata as well as different kinds of abstractions, aggregations or derivatives depending on object characteristics. Examples of such abstractions/aggregations/derivatives are tables of contents and indexes, full text index items, thumbnails, music and video abstractions (e.g. colour histograms or shape abstractions) and signatures. Surrogate metadata records as part of these surrogates are sets of RDF triples.

These atomistic surrogates can be linked to each other to form complex aggregations, which in turn can be organised as Description Sets based on the DCMI Abstract Model[9] or OAI-ORE Resource Maps[10]. These surrogate aggregations correspond to compound logical objects on the content provider's side such as scanned books or multipart multimedia objects, to give just two examples. The central (and mandatory) element of each surrogate aggregation (everything within the light blue circle in the diagram below) is an aggregation root element with a URI of its own and containing some elementary technical and (mandatory!) licensing information. A 'landing page' rendition of this root element is used to expose Europeana DL content to external software agents such as search engines.

There should be a one-to-one correspondence between remote object entities and internal surrogate entities as well as between remote compound logical object entities and complex aggregations of Europeana surrogates. In such a perspective, the decision regarding the actual boundaries of a complex surrogate aggregation largely depends on the way the object providers conceive the entities they want to make accessible via the Europeana surrogate space.

Furthermore, Europeana surrogates as well as surrogate aggregations will systematically be linked to semantic resources representing concepts as well as to external reference resources representing reference entities such as persons, places and periods in time. Links to these reference resources are used in order to create context for the Europeana surrogates. The reference resources may be part of the Europeana data space or external to it: they are referred to as web resources using a URI in either case.

In either case, the semantic resources surrogates are linked to will be organised as semantic web ontologies (hitherto referenced as 'ontologies'), containing the vocabularies for describing the meaning of surrogate aggregations. Semantic ontologies include thesauri, classification schemes, subject heading systems, taxonomies, and the like. Semantic ontologies will be used to define entities (which

---

[9] DCMI Abstract Model: http://dublincore.org/documents/abstract-model/

[10] OAI-ORE – Open Archives Initiative Object Exchange and Reuse: http://www.openarchives.org/ore/

mostly have a lexical counterpart) both at the concept/class level (by defining the entity classes and the relations between them), and at the word/object level (by defining the allowed instances of semantic ontology classes). The latter mechanism will allow to model authority files in the sense of collections of valid instances. Moreover, domain knowledge such as historical events or biographical information is also modelled via semantic ontologies; a notable example of this is the CIDOC CRM[11].

Semantic ontology classes and objects will be associated to surrogate aggregations in two ways:

(1) implicitly, as string metadata attributes, such as the dc:subject attribute connecting an aggregation to a term from a classification scheme, or the dc:creator attribute connecting an aggregation to an entry in an authority file;

(2) explicitly, via *classification* association links to web resources (URIs). An example of a classification association is the "is about" association, relating an aggregation to a topic (i.e., class in a semantic ontology); another example is the "represents" association, relating an aggregation (or a single surrogate) to an object, instance of a person or an event class.

4.1.1.6 Associations

Both the links within aggregations (part-whole relations) and between aggregations as well as between surrogates/aggregations and reference resources are not yet given definitive types in this version of the specification document but the need of more 'specialisation' is evident. This 'specialisation' is likely to draw upon the Resource and Content domains of the DELOS reference model[12], standardisation attempts such as MPEG21 DIDL[13] or PRISM[14], as well as on the ORE Abstract Model and the related vocabulary as part of the ORE specifications[15]. Another promising starting point for typing relations is the list of FRBR property declarations as part of FRBRoo [16] as well as the CIDOC CRM properties referred to in this document. The result of the work on typing relationships/links will be a framework for expressing complex multimedia object structures as well as structural relations between objects and reference entities. An ontology (or several ontologies) of such structural relations - also known as a content model – will be needed in this respect.

However, we can assume some initial guiding lines and distinguish the following association types:

- *content* associations, relating a surrogate to other surrogates, to reflect structural relationships between the corresponding objects. These associations can be further characterized as:
    - associations defining object structures; different content models will have different types, but they can be taxonomized as specializations of the IsPartOf relation;

---

11 CIDOC CRM – Conceptual Reference Model: http://cidoc.ics.forth.gr/

12 DELOS Reference Model: http://www.delos.info/ReferenceModel

13 MPEG 21 DIDL: http://xml.coverpages.org/MPEG21-WG-11-N3971-200103.pdf

14 PRISM, Publishing Requirements for Industry Standard Metadata, http://www.prismstandard.org/

15 OAI-ORE Abstract Data Model: http://www.openarchives.org/ore/0.1/datamodel

16 http://cidoc.ics.forth.gr/docs/frbr_oo/frbr_docs/FRBR_oo_V0.9.pdf

- associations capturing versioning; there are several models which can be used for inspiration, depending how sophisticate the underlying mechanisms need to be; versions can form a single line or a tree or a directed acyclic graph;
- the FRBR associations.

- *description* association*,* relating surrogates to the metadata objects describing them in some description ontology. The singular here means that in principle one should not have more than one association; but if needed, these associations may form a taxonomy with one specific association being the root;
- *naming* associations, relating surrogates to their appellations, that is object-level elements of terminological ontologies. There is an implicit associative mechanism here, because if x is the name of a surrogate and y is a synonym of x, then also y can be used as a name for that surrogate;
- *classification* associations, relating surrogates to the concept-level elements of terminological ontologies. An example is the "is about" association, relating an surrogate to a topic; another example is the "represents" association, relating an surrogate (or a portion thereof) to a world entity representation; a third example is the "instance of" association, relating a surrogate to a class, like Monna Lisa being an instance of Renaissance Art. Inference plays a major role also here, in the sense that concepts in terminological ontologies may be connected by logical relations (sumbsumption, equivalence), which therefore apply to the relatum. A classical example is the inference that has as antecedents "object X is an instance of Renaissance Art", "Renaissance Art is Art" and as consequent "object X is an instance of Art". Different associations may have different logical properties, which should be stated by specifying the semantics of the associations.
- *similarity* associations, relating the surrogate of an object to the surrogates of the objects that are similar to it. Similarity can be defined along several axes:
  - *content-based similarity,* capturing resemblance between text (as established by information retrieval models), images, audio, video and audio-visual objects. These associations are typically computed on demand rather than stored. An important point is that this kind of associations are typically application-dependent, so their set should be extensible.

recommendation, capturing resemblance between objects as established by experts (possibly via annotations), usage (people who accessed one object often accessed the other one), or other criteria.

In this respect, it remains to be determined whether Europeana will fundamentally distinguish relations within an aggregation from those linking aggregations to each other or to reference resources. This issue as well as the potential internal recursive structure of Europeana surrogate aggregations has a counterpart in the OAI-ORE regarding the distinction of internal and external relations. While abstracting from the wealth of object modelling options and choosing a few, general ones that capture structural relations in the most popular existing standards Europeana should take good care to evolve in line with the ORE model in order to preserve the interoperability potential with the repository community.

Europeana object surrogates can thus be simple entities or can be aggregated into potentially complex logical entities and related to other surrogates and reference resources. A logical overview of this is given in Figure 3 hereafter:

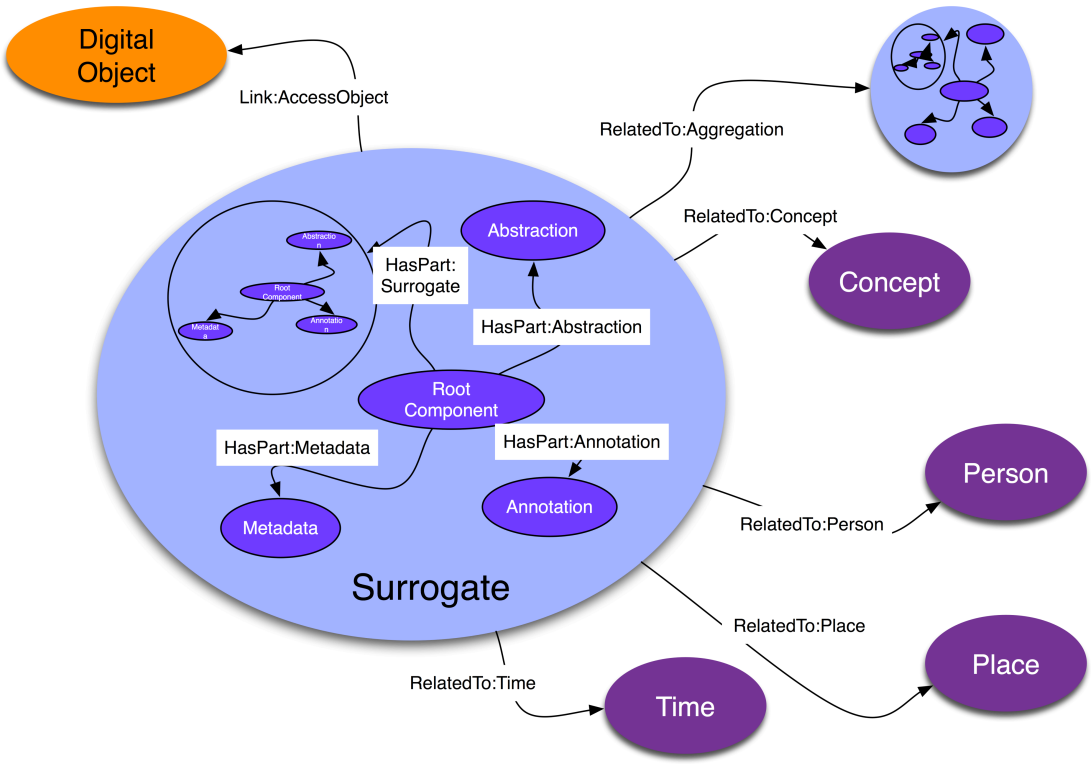

Figure 3: Surrogate Model Logical Overview

Surrogates will have, metadata, abstractions (such as tables of content or colour histograms) and annotations as parts, surrogate aggregations additionally have component surrogates which in turn may have a complex internal structure as described above. The 'HasPart' link is used to point from the surrogate to its components. The 'RelatedTo' link is used to point from the surrogate to external entities. As said before, both links evidently need to be specialised and 'typed', and one important task in further working out the functional specifications of Europeana is to produce a list of values specifying the type of relation (work to be done jointly with the ORE initiative).

Surrogates can be exposed via the Europeana API and/or Europeana portal services, but the API should be underlying the portal, too, and exposure via API should thus be considered the standard way of surrogate delivery.

An important consideration for the surrogate model is that it is necessary that surrogates can be referenced.

The degree of object granularity to be delivered is determined by the content provider who additionally should be given the possibility to indicate the object modelling schema he is referring to in conceiving object building blocks (e. g. TEI[17] or DocBook[18] or MPEG21[19]). This information will be used by Europeana when creating surrogate representations of these objects for translating the relations the content provider has

17 Text Encoding initiative TEI) - http://www.tei-c.org/Guidelines/P5/

18 DocBook - http://www.docbook.org/

19 MPEG21 - http://mpeg-21.itec.uni-klu.ac.at/cocoon/mpeg21/

conceived on object level to the structural relations known within the Europeana content model (cf. above).

4.1.2 Europeana Metadata Requirements

4.1.2.1 Object metadata

The basic OAI-PMH mechanism may be used to harvest simple Dublin Core metadata from the content providers [20]. It is foreseen that Europeana will also receive additional object-specific metadata, either through the OAI-PMH getRecord request with appropriate metadataPrefix or through other means. This more detailed metadata should be delivered according to an XML format that is agreed between the content provider and Europeana management. Possible formats include: qualified Dublin Core conforming to an Application Profile such as the one defined for TEL[21], METS[22], EAD[23], EBU Core[24], Immix[25], CIDOC CRM, MODS[26], MARCXML[27], MPEG-21 . CDWA[28], Dismarc[29], museumdat[30] and Moreq2[31]. The XML schemas for these metadata sets need to be provided by the content provider.

All incoming metadata in one of the agreed formats need to be converted to a common internal format, the semantics of which are described in the table below. The table is intended to include the full list of metadata elements that will be understood by Europeana. In addition, other metadata, if supplied by the data providers, can possibly be used for full-text indexing.

The link with the metadata format listed in section 4.2 of D2.2 is that some of the metadata elements below may be derived from Dublin Core metadata that is delivered as part of the initial OAI-PMH harvesting mechanism. As the default format for OAI-PMH is just simple Dublin Core, the metadata received through that mechanism will not meet all of the more detailed requirements outlined below. Therefore, additional object-specific metadata will also be harvested from providers if available.

---

20 Dublin Core Metadata Element Set, version 1.1: http://dublincore.org/documents/dces/

21 TEL Application Profile for Object: http://www.theeuropeanlibrary.org/handbook/Metadata/tel_ap.html

22 Metadata Encoding & Transmission Standard (METS) - http://www.loc.gov/standards/mets/

23 EAD – Encoded Archival Description: http://www.loc.gov/ead/

24 EBU Core Metadata Set: http://www.ebu.ch/metadata/documentation/EBUCore/tec_doc_t3293_2008_FinalDraft.pdf

25 iMMix, Nederlands Instituut voor Beeld en Geluid, contact Annemieke de Jong adjong@beeldengeluid.nl

26 Metadata Object Description Schema (MODS) - http://www.loc.gov/standards/mods/

27 MARC 21 XML Schema - http://www.loc.gov/standards/marcxml/

28 http://www.getty.edu/research/conducting_research/standards/cdwa/

29 DISMARC – Discovering Music Archives: http://www.dismarc.org/

30 http://museum.zib.de/museumdat/

31 Moreq2 – Model Requirements for the Management of Electronic Records: http://www.moreq2.eu/

A particular case where the object-specific metadata is necessary is the case of what we refer to as 'complex' objects. For these objects, structured metadata needs to be available as well that contains a description of a coherent collection of objects that need to be seen in context of the collection.

While for 'simple', 'atomic' objects, the processing of incoming metadata can be a more or less straight-forward conversion from the XML format provided to the internal metadata format, in the case of 'complex' objects, received metadata will need to go through a more elaborate process. Surrogates need to be generated for each of the components of the object and the metadata need to be decomposed into metadata records for the individual components. Appropriate linking between the surrogate for the 'root' object and the component surrogates and between the component surrogates, is necessary to precisely reflect the internal structure of the object.

The metadata elements described in the table need to be seen from a purely functional and semantic perspective. There is no pre-defined mapping to any particular implementation or metadata standard, although the Dublin Core properties are used as examples. The actual encoding of this internal format is left to the implementers.

Of the elements listed in the table only the first four are mandatory (location, owner, format and rights). For the other elements, as many as possible and relevant should be made available. These metadata can be derived from OAI-PMH metadata (if that is the mechanism used by a particular provider), from specific metadata that the provider is able to supply or from manual intervention by either experts or end-users. The metadata can be enhanced in various ways, harmonizing and/or linking to controlled vocabularies or authority files.

Semantic description	*Source – comment*	*Example DC property*
Location of object (mandatory)	Expressed as a URI. For simple objects, from dc:identifier in OAI-PMH metadata and/or from specific metadata; for complex objects, pointers to the individual components are derived from further processing of detailed metadata; to be used for linking to the original object	dc:identifier
Institution holding the object (mandatory)	From dc:source in OAI-PMH metadata and/or from specific metadata; incoming data needs to be in standardised form so it can be converted to link to the record in the Europeana provider name authority file; to be used for institution-based searching and for grouping results by institution	dc:source
Object format (mandatory)	From dc:format in OAI-PMH metadata or from specific metadata needs to be one of the	dc:format

Semantic description	*Source – comment*	*Example DC property*
	file types supported by Europeana; to be used for format-based searching and narrowing of results	
Rights	From dc:rights in OAI-PMH metadata and/or from specific metadata should be either a term of Europeana terms-of-use vocabulary or a Creative Commons[32] license; to be used in presenting usage restictions to the user	dc:rights
Contributor	From specific metadata; possibly enhanced through automatic processing, linking to name authority file, manual enhancement by experts; to be used for simple search and name-based searching	dc:creator and dc:contributor, including refinements
Creation date	From specific metadata; to be used for narrowing results	dcterms:created
Description	From specific metadata; possibly in multiple languages; to be used for simple search	dc:description
Geographic coverage	From specific metadata; automatic enhancement; possible further manual enhancement by experts; to be used in map-based searching and presentation	dcterms:spatial or dc:coverage.spatial
Language	From dc:language in OAI-PMH metadata; to be further processed and harmonized using ISO 639[33]; to be used to restrict simple searches to specific language or narrowing results	dc:language
Modification date	From specific metadata; date of last update; to be used for narrowing results	dcterms:modified

[32] Creative Commons: http://creativecommons.org/

[33] ISO 639-3 – Codes for the representation of names of languages: http://www.sil.org/iso639-3/

Semantic description	*Source – comment*	*Example DC property*
Object type	From dc:type in OAI-PMH metadata and/or from specific metadata; possibly enhanced by particular genre (e.g. painting, book, video) or domain (library, museum etc.) or theme (an initial list as defined for the prototype) that could be derived from detailed metadata or added by further processing or manual enhancement by experts; to be used for relevance ranking and for narrowing of results	dc:type
Publication date	From specific metadata; to be used for narrowing results	dcterms:issued
Publisher	From dc:publisher in OAI-PMH metadata and/or from specific metadata; possibly enhanced and harmonized; to be used for name-based searches	dc:publisher
Relation	From specific metadata and from processing of 'çomplex' object metadata to create a network of surrogates reflecting the structure of the 'complex' object; needs to be further analysed to define the appropriate set of relation types	dc:relation and various standard and local refinements
Subject	From dc:subject in OAI-PMH metadata and/or from specific metadata; processing to link to 'semantic nodes'; possible enhancement through manual enhancement by experts and end-users; to be used in simple search and subject-specific search	dc:subject
Temporal coverage	From specific metadata; automatic enhancement; possible further manual enhancement by experts; to be used for time-line presentation and navigation	dcterms:temporal or dc:coverage.temporal
Title	From dc:title in OAI-PMH metadata and/or from specific metadata; any formal, informal, abbreviated or parallel title should	dc:title

Semantic description	*Source – comment*	*Example DC property*
	be included; to be used in simple search and subject-specific search	

For every metadata statement, it needs to be recorded, as a minimum, (a) when it was last modified and (b) who made the modification (system, expert, user) in order for the system to be able to assign weights to the values depending on the trustworthiness of the metadata and the particular use that is made of the metadata (e.g. to allow for the functional requirement that user-provided metadata has a higher value than expert-provided metadata).

4.1.2.2 User metadata

Property	Comment
User ID	Internal, unique identifier
User status	Anonymous/registered
Last access	Date and time that user was last in the system (from cookie if user is anonymous, from login for registered users). May also be used to delete user records after certain period of inactivity.
Cookie info	
Login name/password	As provided by user when registering; password encrypted
User type	General/expert (to be specified by user when registering)
User domain	Archive/Museum/Library/Audiovisual Archive/Other (to be specified by user when registering)
User interest profile	Any type of information that the registered user wants to provide and that can help better ranking and suggestions. Possibly also generated by the system on the basis of past behaviour (also for returning anonymous users)
User subscriptions	Indicating which Europeana Communities the user is a member of (for registered users only)
User space	Pointer to private space assigned to user (for registered users only)
User e-mail address	If provided by user (for registered users only)

4.1.2.3 Provider metadata

Property	Comment
Provider ID	Internal, unique identifier

Property	Comment
Provider status	Content holder/Aggregator
Provider domain	Archive/Museum/Library/Audiovisual archive/ Archaeology/Monuments/Other
Last harvest	Date and time of last successful harvesting
Harvesting mechanism	OAI-PMH/FTP/etc. (code for any supported mechanism)
Harvesting format	Available metadata format for harvesting
Harvest address	URL where file will be available for harvesting
Provider Name	As supplied by provider
Address information	
Web site address	
Contact person name	
Contact e-mail	
Aggregator link	Provider ID of Aggregator if content from this provider is received from an aggregator
Content holder link	(Repeatable) Provider ID of Content holder that this Aggregator aggregates

4.1.3 Queries

Europeana will host a large volume of data, including content and metadata on a large number of digital objects, semantic structures capturing lexical and domain knowledge, user profiles, usage logs and statistics, and other. These data will be modelled according to the Europeana Data Model, will be managed in secondary storage and will be accessed via a querying mechanism. A querying mechanism consists of a query language and a query evaluation engine, returning the result of queries according to the semantics of the query language. Europeana will make the querying mechanism available through a programmatic interface, which will be used by the applications serving the different types of users. Based on the User Groups of Europeana outlined in Section 3.1, the querying mechanism will serve:

- *end users,* wishing to discover objects relevant to their information needs or to browse the Europeana information space; these users typically sit behind a graphical user interface (such as the Europeana portal), which acts as a mediator between the end user and Europeana;
- *external applications,* aiming at extracting from Europeana the information required for carrying out a specific application task or for integration with other information; Web search engine crawlers are typical external applications;

another type of applications will be those serving meta users will want to access logs and statistical information to do performance analysis;

The *internal components* of Europeana, will also use queries to obtain the information required for implementing a Europeana service such as creating an output page, administering the system or supporting a certain browsing of the digital library.

All these actors will formulate their information needs as expressions of the query language describing the required information. Europeana will evaluate such queries in a way transparent to the user, and will return the desired results.

The query language is therefore a fundamental tool For Europeana. Based on the user requirements, the following query types will have to be supported:

- content-based queries on multimedia documents; a content-based query is typically a prototype of the desired object or of one of its features (in the case of images, a feature could be the color distribution or a shape); in response to a content-based query, Europeana will produce a rank of the stored objects, that is an estimation of the degree to which each object is similar to (matches, resembles, approximates) the given query prototype. The addressed media types can be texts, still images, audio objects or video objects. The similarity criteria may be application-dependent, therefore it will be highly desirable that Europeana will provide basic similarity predicates and will give to the application designers the possibility to extend these basic predicates with application-dependent predicates, capturing application-dependent similarity criteria. Similarity search can be directly used by users, to perform query by example, or can be used behind the scenes by classification algorithms [LeSaux_et_al2004] and semantic search techniques [Amato_et_al2007] to factorize their complex tasks.
    - as a special type of content-based queries, Europeana will provide *simple search queries*, that is full-text queries, evaluated according to a classical information retrieval method against textual objects or metadata records interpreted as texts
- semantic-based queries on metadata; these are expressions of a logical language, including Boolean operators, variables, attributes and associations. The evaluation of such queries may require reasoning, if lexical or domain knowledge structures are involved. There are two special types of semantic-based queries:
    - *fielded search queries*, consisting of conjunctions of simple conditions on attribute values of flat objects (e.g type = "image") or metadata records (e.g. dc:creator = "john");
    - *ontology queries,* consisting on expressions ranging over the contructs of ontologies, and allowing to extract information concerning the concepts and the relations defined by the ontology (e.g. "the terms which are synonymous with bank", or "the sub-classes of painting").

  While semantic-based queries are exact match queries, that is an object either satisfies or does not satisfy a query, the result of such queries should nevertheless be a ranking of the objects, like in the case of best match queries, in order to make large query results easy to consume for the requester. In order to derive this ranking, the underlying engine should exploit all the available information on the entities involved in the discovery process, that is the query, the user, the object and the context in which the process takes place. Ranking can also be done according to partial matching of query parts (if more than one field / quality is requested). One important piece of information

is the source of the metadata, which must be recorded by the Europeana data model and used to determine the rank of the associated object. Such usage must also consider the domain which the object comes from, since the same attribute may have a different selectivity for different domains (e.g. dc:title is much more selective for books than for paintings). A special treatment should be given to user-contributed tags. Depending on who the user is such tags should have more importance than expert-contributed metadata. While the actual ranking will be the task of the retrieval machine, the query language will have to specify appropriate sorting criteria.

It should be noted that the query language is paramount to define an information discovery facility adequate to the expectations of the above mentioned Europeana actors; thus, its basic characteristic should be *completeness* with respect to the European Data Model, meaning that queries must be expressive enough to denote all the information representable in the model, thereby providing access to *all* the stored information. This requirement should not be traded with usability, which is a user interface requirement and must therefore be considered upon designing the interaction between the user interface itself and the user.

Current technologies offer a wide range of choice regarding the query language and the underlying query evaluation engine. For content-based queries, decades of research on multimedia information retrieval have created a solid technology, allowing to perform similarity-based search on millions of objects with impressive efficiency and reasonable effectiveness, especially for text (see below on multimedia indexing). For semantic-based queries, Semantic Web languages offer a widely accepted tool for representing and reasoning about semantic information.

4.1.4 Index Resources

An Index Resource is a data structure managed by an Index Manager to speed up data access, typically (but not exclusively) upon query evaluation.

As outlined in the Queries Section, Europeana will be endowed with a powerful query language serving a very rich Data Model, used to represent an expected huge amount of diverse data; in addition, Europeana will face a very large number of users or applications, simultaneously accessing its information space. In order to achieve an efficient data access under these very demanding circumstances, a very careful design of the Europeana index resources is required, so to guarantee optimal response time. In particular, different index resource types will be needed in order to accomplish the Index Service functionalities. In the following we give a brief description of the foreseeable types of the Europeana Index Resources, based on the previous discussion on queries. We can classify indexes as follows:

- index for content-based queries. Techniques for effective and efficient text retrieval have been studied since many decades and are nowadays adopted by many tools widely used to search for textual information in large scale repositories. Recently also techniques for content-based retrieval of multimedia data have filled the gap that prevented them to be used on a large scale. For these types of data, it is harder to achieve the same level of efficiency as text, due to the fact that queries are usually non-sparse vectors. Yet, there exist several technologies that can be exploited. Some techniques were also subject to a standardization process. An example is given by the MPEG-7 visual descriptors that can be used to characterize many visual aspects of pictures and videos. In addition, techniques (data structures and algorithms) for efficient

similarity search on large document sets are available as well. After pioneering works proposing techniques able to deal with data represented in vector spaces [Guttmann1984] and metric spaces [Ciaccia_et_al1997], several enhancements were developed which improved performance of orders of magnitude, for instance, by addressing the issue of approximate similarity search [Amato_et_al2003], or the use of distributed algorithms [Falchi_et_al2007]. Extensive surveys discussing these techniques can be found in [Zezula_et_al2006] as well as in [Samet2006].

- index for queries based on attribute-value pairs (i.e advanced search queries), called *formatted index*. This kind of index is used by database management systems to speed up the evaluation of particular kinds of queries (usually by internally generating and storing redundant information for quicker location of table entries). The choice of the metadata record fields to index (physical database design) is paramount in order to obtain optimal performances (e.g. from the maquette: subject, format, time, place, etc are candidate fields for this index in Europeana). The Who & What, Where and When as well as the social tag based queries will be evaluated by relying on this kind of index.
    - as a special case of a formatted index, we have the index for queries based on keywords from a controlled vocabulary, called *keyword index*. This is a special case of the previous defined index because these keywords usually are given as values of a metadata field (e.g. dc:subject). An important difference between a simple formatted index and a keyword index is that the values indexed by the latter may be terms arranged in a taxonomy capturing a specialization/generalization relation, or lemmas in a thesaurus, offering a rich set of relations between them. In both cases, the relations existing between the keywords must be taken into account during the query evaluation process.
- index for navigating the associations between objects, called *navigation index*. This kind of index are introduced to enhance performances in retrieving stored associations among objects surrogates, such as: isComponentOf, IsPartOf, IsRelatedTo, and IsVersionOf. These associations needs to be navigated upon displaying query results, or building landing pages.

For scalability and performance reasons, partitioning each one of the presented index by some properties (e.g. location, language, etc.) should eventually be considered.

For each type of index, the following functions need to be offered by Europeana:

- *Create index*. This function is used to perform the initial creation of an index on the values of an object attribute (such as a metadata field), which may be simple or compound (*i.e.* a concatenation of two or more values). A name is associated to each index at creation time, and later used to identify it. Along with value/object pairs an index contains also statistics about its content (e.g. number and distribution of distinct values). These statistics are used for query optimization purposes.
- *Delete index*. Removes an index.
- *Add Entry Value.* This function is used to add an entry in an existing index. An index entry consists of a data structure that maintains for each distinct value (simple or compound) a set of pointers to the objects having that attribute

value. A new entry is created when the first pair is added for a given value and updated when a new object identifier is supplied for the same value.

- *Remove Entry Value*. Removes the given object identifier from the entry of the given value. When the last object identifier is removed from an entry, the index entry is removed too.
- *Build Index*. This function performs a batch creation of an index for a given attribute. Usually this operation follows an harvesting phase, and is used instead of the add entry function in order to enhance performance.
- *Rebuild Index*. It is used to re-organize an index when a large amount of add/remove entry operations have been performed which could lower the index performances.
- *Index lookup*. It is used to extract information from an index. There are two kinds of lookups:
    - *value lookup*, this is typically a range query which, given a value interval, returns the objects whose attribute value falls into the given interval;
    - *statistics lookup*, used by the query optimizer to generate an adequate access plan for a given user query.

It should be noted that most of the above functionalities are supplied by existing software components, usually embedded in query evaluation engines of database management systems.

Additionally, it should be clear that query language and index structure need to be compatible: the most sophisticated query language can't do much with a single-field index. Additionally, the search engine needs to be able to produce rankings for different specified profiles according to object types, query characteristics, etc…

4.2 Functional Model Overview

4.2.1 Introduction

The Functional Model describes the tasks that Europeana must be able to perform in order to satisfy the expectations of its users. These tasks are grouped together in areas, called functional areas; each functional area collects tasks semantically related and can be thought of as representing a macro-functionality, further analyzed in tasks of finer granularity.

The functional areas of Europeana have been defined as a result of the analysis of user requirements and are summarized in the following list:

- the Capture and Dissemination area, offering functionalities for populating Europeana and disseminating its contents;
- the Object Management area, providing functionalities to manage digital content, the corresponding surrogates and the associated metadata;
- the Discovery area, supporting the indexing and searching of the Europeana content according to several paradigms;
- the User area, supporting the functionality for managing users, from single persons to institutions and dynamic communities;

- the Access area, supporting access to both Europeana services and content as well as to external resources.

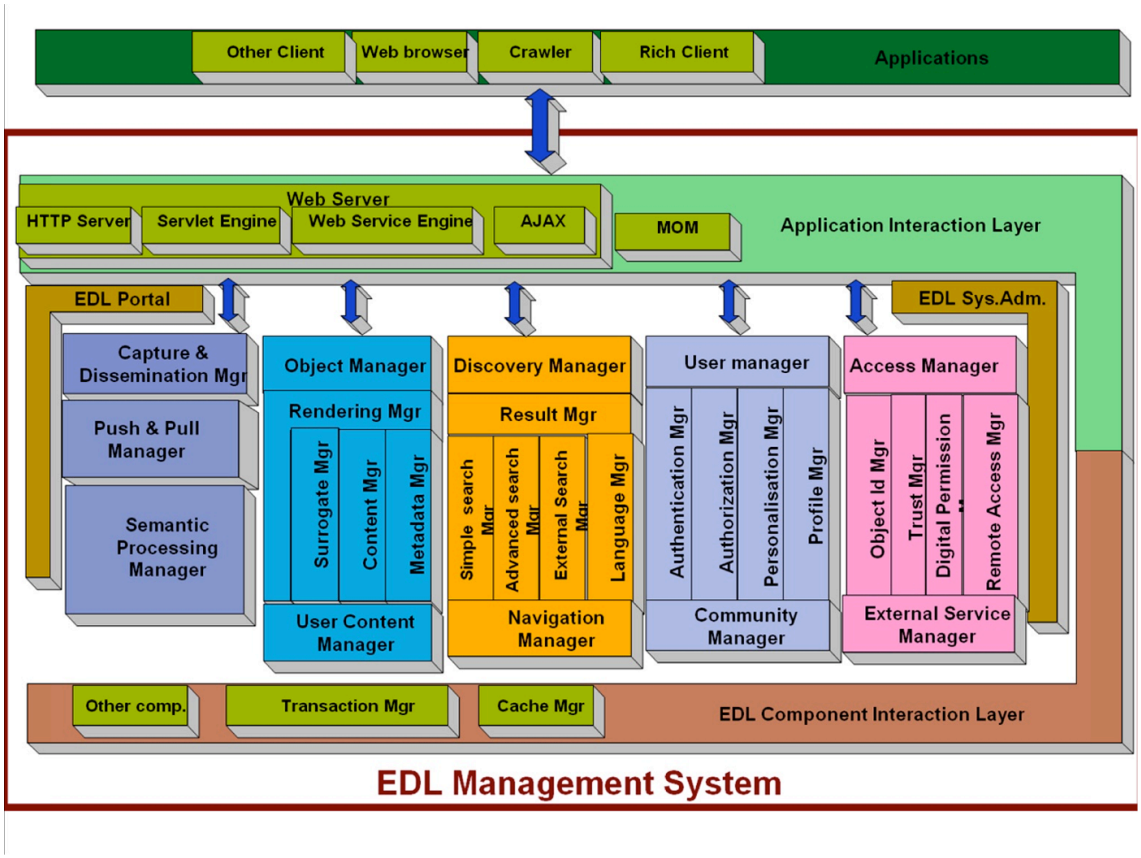

Figure 4: Europeana Architecture

The above figure sketches the main functional actors in the Europeana landscape. The five areas illustrated above are shown in the diagram as the main blocks within the Europeana system. Each block consists of several boxes of the same colour: the top box of each block reflects the functional area, whereas the other boxes analyzes the area in components of a finer granularity addressing a finer functional area within the block. In more technical terms, the top box of each block can be seen as a *interface* implemented by the components below it.

Interaction with external applications (Application Interaction Layer) will be managed by a set of special components, built by embedding existing, mainly open source, communication frameworks; a few of these components are highlighted in the figure as a way of showing the nature of the Application Interaction Layer.

At first stage, interactions with external applications will be based on the family of protocols built on HTTP (e.g. SRU[34], SOAP[35], XML-RPC[36] etc.), however the Interaction Layer will be implemented in a modular way to enable system administrators to add components supporting other communication protocols.

The back end of the Europeana will be implemented by components (grouped in the Component Interaction Layer) that will provide functionalities for system integration.

---

34 SRU – Search/Retrieval via URL: http://www.loc.gov/standards/sru/

35 SOAP – Simple Object Access Protocol: http://www.w3.org/TR/soap/

36 Extensible Markup Language Remote Procedure Call - http://www.xmlrpc.com/

A description of the components implementing the functional areas will be presented in the following, including how they interact with other components, and, where appropriate, the candidate technology or standard for the implementation.

4.2.2 Capture and Dissemination

Components:

- the Push and Pull manager which gets content from and delivers content to content providers, including metadata, annotations, user collections etc.
- the Semantic Processing Manager that transforms, if necessary, the content according to the Europeana vocabularies. It performs validation, normalisation, transformation, mapping of metadata and may need to interact with various kinds of external resources (schemas[37], ontologies[38], Wikipedia), provided these have a coherent structure and syntax (preferably expressed in XML[39]) and with associated descriptive information (schema, DTD[40] or similar).

4.2.3 Object management

Components:

- The Content Manager is responsible for analysing and pre-processing the digital content, mapping the incoming content to Europeana data structures and preparations for the Surrogate Manager. Because of the great heterogeneity that is to be expected on content, the content manager must be able to understand any content model. The solution may be based on the content model underlying the Java Content Repository API[41], which is becoming a de facto standard for content repositories.
- The Metadata Manager is responsible for managing metadata at all levels: ontologies, schemas and records. Semantic Web[42] technologies should be used for interoperability and generality. Several implementations exist, and in particular the Jena Semantic Web Framework implementation[43] seems a very viable option. The BRICKS[44] Metadata Manager can then be considered as a candidate implementation.
- The Surrogate Manager is responsible for supporting the creation of surrogates, including the Surrogate Root Component, as described in chapter 4.1.2, as well as all functionalities that each surrogate makes available. From the surrogates, various variants of presentations can be generated, including 'landing pages' for display and for consumption by search engines, either by presenting these pages on-the-fly or by pre-compiling them.

---

37 Schema: http://en.wikipedia.org/wiki/XML_schema

38 Ontology: http://en.wikipedia.org/wiki/Ontology_(computer_science)

39 XML – Extensible Markup Language: http://www.w3.org/XML/

40 DTD – Document Type Definition: http://en.wikipedia.org/wiki/Document_Type_Definition

41 Java Content Repository API (JSR-170): http://jcp.org/en/jsr/detail?id=170 and JSR 283: http://jcp.org/en/jsr/detail?id=283

42 Semantic Web: http://www.w3.org/2001/sw/

43 Jena Semantic Web Framework: http://jena.sourceforge.net/

44 BRICKS project: http://www.brickscommunity.org/

- The Rendering Manager: responsible for creating output files, depending also on the device where the files are to be delivered. It may also provide links to additional information, such as advertising (e.g. events related to user profile or to results), Google Maps[45] and Wikipedia when this is relevant to the object selected. It will get XML streams from other components and produce output depending on the channel (e.g. PC mobile device). This includes preparation of content for use on accessibility devices, such as screen readers, Braille displays, etc.
- The User Content Manager is responsible for managing the user content within Europeana, either for single-user or collaborative spaces. This includes the management of virtual collections, baskets with selected items, folksonomies[46] and the like.

4.2.4 Discovery

Assuming users would find useful both a simple (Google-like) and an advanced (database-like) search, the discovery of information will be supported by the following components:

- The Simple search Manager is the component responsible for scoring simple queries, possibly after expanding them for crossing language or other semantic boundaries. It will be implemented using standard software packages such as Lucene[47]. For personalising the object ranking on the basis of user preferences, the Personalisation Manager may be involved.
- The Advanced search Manager is the component responsible for scoring advanced queries, involving attribute values, comparison operators and Boolean connectives. An important issue here is the query language that will be supported. A possible choice, consistent with the above suggestion of using Semantic Web languages for metadata, is to use an RDF[48] query language, e.g. SPARQL[49] which is supported by Jena Framework. The BRICKS Query Mediator can be used as a basis for implementing both simple and advance search. Advanced search must be done on a common set of fields and will include:
    - Similarity search
    - Semantic discovery methods including faceted browsing.
- The Search gateway is responsible for reaching on-the-fly (i.e. at query evaluation time) sources which are not harvestable.
- The Language Manager is will handle all functions related to Language, such as query and result translation, management of user-defined dictionaries and the like. It will link to external resources, such as dictionaries, thesauri and translators.
- The Result Manager implements all functionalities offered on the result of a discovery, including query refinement and relevance feedback. It may allow the separate display of results of queries evaluated against external resources, such as Wikipedia.

---

45 Google Maps: http://maps.google.com/

46 Folksonomies: http://en.wikipedia.org/wiki/Folksonomy

47 Apache Lucene: http://lucene.apache.org/java/docs/

48 RDF – Resource Description Framework: http://www.w3.org/RDF/

49 SPARQL: http://www.w3.org/TR/rdf-sparql-query/

- The Navigation Manager manages user interaction sessions, keeping track of the history of visited pages, possibly including external pages. Exploring the Europeana information space by content similarity can also be supported.

4.2.5 User

Components:

- The Authentication Manager is responsible for authenticating users, implementing single sign-on and integrating with the European infrastructure under development by Terena[50].
- The Authorisation Manager is responsible for managing users permissions.
- The Profile Manager is responsible for managing user models, including preferences
- The Personalisation Manager will use preferences to personalise content for a user, e.g. on the results of search operations, or on browsing.
- The Community Manager will support communities of users created for purposes of collaborative work.

4.2.6 Access

Components:

- The Object Identity Manager supports the creation of identifiers, the mapping to external identifiers, and the involved resolution process.
- The Trust Manager manages the trust between Europeana and the external sources with which Europeana must interact for various purposes (SAML[51] standard based models such as Shibboleth[52] and Liberty[53] may be adopted for this functionality) This component needs a list of external resources (e.g. providers) and external services and collections
- The Digital Rights Manager
- The Remote Access Manager manages access to digital objects: for instance if reaching the object requires a complex interaction, it is the responsibility of the Remote Access Manager to support this interaction.
- The External Service Manager manages the interaction with external service providers.

5 Individual Functional Areas

5.1 End User Use Cases

5.1.1 Interaction of Europeana with End Users

Either in the search or browse screen, the user is presented with the latest 20 surrogates added to the portal. Besides, (s)he is presented with a widget containing a

---

50 TERENA - Trans-European Research and Education Networking Association: http://www.terena.org/

51 SAML – Security Assertion Markup Language: http://en.wikipedia.org/wiki/SAML

52 Shibboleth: http://shibboleth.internet2.edu/ 12

53 Liberty Alliance Project: http://www.projectliberty.org/ 13

check-box list with the major object types, i.e. is offered a faceted browsing of the latest 20 surrogates of any object type (or combination of object types):

√	text (n objects)
√	image (m objects)
√	music (p objects)
√	speech (q objects)
√	video (r objects)

5.1.1.1 Find a known object

In this scenario, the end-user is looking for an object (s)he has in mind. He will start the search process supplying a known property of the object, such as title or creator. During the typing of the keyword/syntagm (s)he is presented with the set of index terms having as prefix, the supplied string. Each index term is accompanied by the number of surrogates it corresponds to. Thus, the user knows in advance how many entries the hit list will have.

The search box has a side bar with a check-box list of the main properties (attributes) of the objects, so the user can select the properties the keyword refers to:

√	title
√	contributor
	...
√	subject

Results:

a) The best case: succesful first attempt. The user is offered the object surrogate.
b) The worst case: failed attempt: the object exists but is not displayed or it doesn't exist at all. In this case, the portal should:
   - check the spelling of the key word(s) used, and suggest other spelling variants;
   - try to suggest similar objects:
     - if there is a more generic term in the knowledge base than the user-supplied keyword/syntagm, the system offers surrogates indexed with that term;
     - if the search expression consists of multiple words, the system offers surrogates indexed with combinations of fewer words;
c) The good case: less than a screenful (say 20) of surrogates are retrieved. The hit list is displayed.

d) The usual case: more than a screenful of surrogates are retrieved. A faceted browsing screen is displayed.

In any case, a widget with the page of the general index where the search word/syntagm has (or should have) its place is displayed, in order to present the user with the collocated words/syntagms.

5.1.1.2 Browse surrogates having a known property (such as subject)

The portal displays:

a) a faceted browsing screen, the first facet being the specific terms (i.e. those related to the supplied keyword via a generic-specific relationship - if such a relationship is present in the knowledge base);
b) the widget with the general index page where the keyword is placed.

5.1.2 Interaction of Europeana with providers / aggregators

Assuming that the elementary requirement of interaction between Europeana and providers of content is consensus regarding the entities to be exchanged, this chapter provides an outline of the Europeana object model as a basis both for interoperation with these content providers and with external co-operating federations such as DRIVER[54] or other content aggregators (e.g. EBU).

After that, specific aspects of Europeana-provider interaction are discussed, and within each section requirements from the content providers are identified in italics.

5.1.2.1 Europeana Object Model

Aggregators and other content providers need to provide identifiers, metadata files, vocabularies in SKOS[55] form, links to semantic nodes, licensing and rights information and access to the original digital objects.

5.1.2.2 Metadata

Any input can be accepted as long as common core attributes as specified are included and XML schemas are provided.

The option to export Europeana content to other systems should be considered.

The common metadata should be provided by the content providers and aggregators in XML. Post-processing to RDF should be done by Europeana.

The mapping from other metadata formats such as MODS[56], qualified Dublin Core, EAD, metadata based on CIDOC CRM and FRBR[57]etc. to the Europeana Dublin Core format should be done by the content providers and aggregators.

5.1.2.3 'Semantic' Issues

The work to turn this in a 'European Ontology' and more specifically the mapping of these concept schemes cannot be done in the context of Europeana alone but must be

---

[54] DRIVER – Digital Repository Infrastructure Vision for European Research: http://www.driver-repository.eu/

[55] SKOS – Simple Knowledge Organisation Systems: http://www.w3.org/2004/02/skos/

[56] MODS – Metadata Object Description Model: http://www.loc.gov/standards/mods/

[57] FRBR – Functional Requirements for Bibliographic Records: http://www.ifla.org/VII/s13/frbr/frbr.pdf

made be part of the wider EC research agenda. However, Europeana will have to contain instruments that can be used to produce such mappings and to promote best practices.

Aggregators and other content providers should supply concept schemes expressed in SKOS.

5.1.2.4 Identifiers/Versioning

Standard identifiers for digital objects are strictly mandatory, and the content aggregators should be responsible for persistent resolution.

Reprocessing of modified object should be triggered by an explicit versioning approach with explicit descriptive information.

Aggregators need to decide what they are identifying, an 'object' or individual digital components.

Europeana will need to provide guidelines and report on good practice in this respect. Europeana should also offer a persistent resolution service with PURL[58] as a likely candidate technology.

5.1.2.5 Rights Management and commercial aspects

Europeana should be seen as an exposure channel for licensed content, while all rights management issues remain with the content providers and aggregators.

Europeana itself should develop an overall policy for rights management and explicitly define what rights are granted and what guarantees are given for use of the object surrogates.

Contracts between content providers, aggregators and Europeana need to specify what Europeana may do with content supplied.

A list of legal and technical options and prerequisites has been prepared by Patrick Pfeiffer and will be submitted to the Foundation for further treatment and forging of an Europeana policy.

5.1.2.6 Ingestion/Delivery Methods

All data transfer will be based on XML structured files (online or on external media).

Not all domains may be equally well served by OAI, especially in relation to data volume and other practical limitations. The use of P2P [59] technologies (BitTorrent[60] and the like) should be considered.

For transfer of the original objects, appropriate mechanisms for packaging and compression need to be selected. Incremental delivery is desirable.

For all objects that are part of Europeana, a URL that points to the object needs to be provided by the aggregators and other content providers.

All components (including sub-components) of a complex entity need to be delivered in a single file as part of one single XML tree.

For all surrogates within its own data space, Europeana needs to supply a URL that persistently points to this surrogate.

---

58 PURL – Persistent Uniform Resource Locator: http://www.purl.org/

59 P2P – Peer-to-peer: http://en.wikipedia.org/wiki/Peer-to-peer

60 BitTorrent: http://www.bittorrent.com/

A Diagram illustrating all workflow aspects of Europeana – provider interaction will be supplied at a later stage.

5.2 Application Interaction Layer

5.2.1 Description

The Application Interaction Layer (AIL for short) includes software components for enabling the interaction between Europeana and external applications with respect to both Europeana services invoking external applications and vice versa.

The rationale behind AIL is that Europeana needs to be able to serve various types of applications, such as clients based on Web browsers (typically, the Europeana portal), fat clients implementing specific GUIs, Web search engine crawlers and applications based on web services[61], to mention a few. On the other hand, Europeana will also need to use external applications for expanding its functionality with relevant services; for instance, search engines focus on different information spaces, language translation services, administration services, and the like. For these reasons, Europeana will expose its whole functionalities through a wide range of interaction paradigms. This will allow and promote the implementation of various applications, including third-party services, adding value to the Europeana information base. Several interaction protocols, both synchronous and asynchronous, should be provided in order to support this vision.

5.2.1.1 Interaction protocols

In the following we briefly describe a set of well known interaction protocols that are recommendable given the present state of the art. More importantly, it is recommended that the Application Interaction Layer of Europeana be extensible in order to allow the inclusion of components not mentioned here or implementing standards not yet fully established.

XML-RPC

This is a standard defined in the late 1990s that allows remote procedure calling over HTTP, using XML for encoding methods invocation ad-hoc responses. There are several implementation of XML-RPC specifications, available in different programming languages. Europeana should support this standard in order to allow the implementation of distributed applications. A typical example of this kind of application is a browser based GUI implemented using AJAX techniques or JSON as a lightweight data-interchange format that developers can use to speed-up their AJAX, as an alternative to pure XML-RPC.

SRU-CQL[62]

SRU is a standard search protocol for Internet, maintained by the Library of Congress, based on XML and using CQL (Contextual Query Language) as a standard syntax for queries. This standard is largely used in the digital library arena, allowing syntactic interoperability among heterogeneous systems. Europeana should implement an SRU server allowing applications to search the digital content by using a specific CQL context set. At the same time Europeana should use this protocol for searching content on providers implementing SRU verbs.

---

[61] Web services: http://en.wikipedia.org/wiki/Web_service

[62] Search/Retrieval via URLhttp://www.loc.gov/standards/sru/cql-bibliographic-searching.html

OAI-PMH

This is a standard defined by the Open Archive Initiative for metadata and content harvesting, providing an application-independent interoperability framework [OAI-PMH]. Europeana should provide an OAI-PMH importer to harvest from external sources and an OAI-PMH server to let itself be harvested. Both these tasks are part of the Capture and Dissemination functional area, described below. Supporting fully automated re-harvesting of OAI data providers is required. The automatic re-harvester should also handle incremental harvesting.

SOAP

This is a well-know protocol used for communication among Web-services. It is XML based and platform neutral. It is usually implemented on the top of HTTP protocol making web services easily accessible. Using this protocol Europeana should provide an API for interacting with the components implementing the main functional areas (i.e. Capture & Dissemination manager, Object manager, Discovery manager, User manager and Access Manager), allowing applications to exploit the whole Europeana functionalities in a language- and platform-independent way. In a number of cases, REST (REpresentational State Transfer Architecture) can be used in place of SOAP for ease of implementation using dynamically generate links in HTML pages. Furthermore, some external services permit only REST integration.

Asynchronous Message Passing

Interaction protocols based on Asynchronous Message Passing model typically allow inter-applications communication by using a middleware agent (e.g. a queue manager), which stores, routes and transforms messages (data items). A protocol implementing this model should be used by Europeana to interact with connectionless applications and to provide caching functionalities to applications accessing Europeana content.

There are many protocols based on this model, mostly strictly tied to a specific programming language or to a specific technological framework (e.g. WebSphere[63], JORAM[64], etc).

5.2.1.2 Supporting development

The AIL should provide external application developers with a set of services to deal with the API exposed by the various system components (i.e. Capture & Dissemination manager, Object manager, Discovery manager, User manager and Access Manager).

Applications using the first three protocols mentioned in the previous paragraphs (i.e. XML-RPC, SRU-CQL and OAI-PMH) can be implemented using existing frameworks and tools in several programming languages. For these protocols. Europeana only needs to support the server side. However, Europena will also need to support the client side of OAI-PMH to be able to harvest OAI servers.

For HTTP based application communication, besides providing a UDDI[65] registry, allowing the discovery and usage of Europeana web services, communication via SOAP and REST together with a well defined API should be provided to support developers.

---

63 http://www-306.ibm.com/software/websphere/

64 http://joram.objectweb.org/

65 Universal Description Discovery and Integration (UDDI) - http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm

This API will have to provide functionalities like discover and access web services, serialisation/de-serialisation, etc.

Finally, due to the lack of standards for the Asynchronous Message Passing model, Europeana cannot provide a universal solution which is language- and platform-independent to applications based on this model. However a java software development kit, based on the well know Java Message Service (JMS[66]) protocol should be provided to java programmers.

Besides, Europeana will need the technical means to provide embedded metadata formats (e.g. COINS) or syndication formats (e.g. RSS, OPML, ATOM) for exporting Europeana content to user environments (e.g blogs).

5.2.2 Constraints and assumptions

Specific attention should be given to security and authentication aspects, when allowing application interaction. This will be discussed in detail in chapter 5.7 (Managing Users).

5.3 Europeana Component Interaction Layer

5.3.1 Description

The Europeana components interaction layer provides a set of facilities allowing system components to interact each other. The role of this layer is twofold: (1) to provide implementation of common functionalities needed by all components, thus avoiding redundancy and inconsistency, and (2) to set the coupling between system components as loose as possible in order to improve scalability and modularity.

5.3.1.1 Components Interaction Facilities

5.3.1.2 Inter-Component communication

The Europeana system consists of several independent software components, some of which could also be distributed throughout the network, which may need to communicate with one another to execute some complex actions. To integrate these components, it is necessary to build an inter-component communication framework implementing synchronous and asynchronous communications. Assuming that Java is a strong candidate to be the adopted programming language for Europeana system development, a set of components implementing well-established java specifications such as RMI[67], JINI[68], JMS[69] etc will be provided.

5.3.1.3 Data Repositories

The Europeana systems will contain several kinds of data as part of surrogates and for management purposes (e.g. binary objects, metadata, user information, indices etc). Data will be managed by specific Europeana components and will be stored in specific repositories. The Components Interaction layer will provide storage and access

---

66 JMS – Java Message Service: http://java.sun.com/products/jms/

67 RMI – Remote Method Invocation: http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp

68 JINI: http://www.jini.org/

69 JMS – Java Message Service: http://java.sun.com/products/jms/

capabilities to all the Europeana components by means of specific subsystems (e.g. Relational Data Base Management Systems (RDBMS), XML based DBMS). This will provide a single persistent layer shared among all Europeana components.

5.3.1.4 Transaction management functionalities

The surrogates managed by Europeana will have, in general, a complex structure, consisting of several related chunks of data whose wholeness must be preserved and guaranteed. The transaction manager is responsible for the Europeana content base integrity and consistency in the following cases:

- when several Europeana system components need to work simultaneously on the same set of data, with immediate updating (concurrent access).
- when executing operations involving a sequence of actions whose atomicity must be guaranteed (ACID transaction).

5.3.1.5 Caching

The caching component will allow Europeana components to store information that can be quickly retrieved. For instance the cache component will be used by the Europeana Result Manager to store query result sets thus providing RDBMS cursor-like functionalities to client applications.

5.3.2 Constraints and assumptions

Specific attention should be given to security and trusting aspects, when allowing Europeana system components distributed over the network to interact with each other. As to interaction with external components, these should have no direct connection to internal Europeana components but communicate via a well defined API providing all necessary functionality and data.

5.4 Capture and Dissemination

5.4.1 Description

Responsibility: harvesting content from contributing institutions (Europeana and / or content aggregators). It will also import the objects from the content providers temporarily for processing by the content manager. In addition, this component exposes methods allowing external applications to harvest data.

It uses:

- the **Push and Pull manager,** which gets links to content from and publish links to content to external providers, including metadata, annotations, user collections
- the **Semantic Processing Manager**, which transforms the external references to content according to the Europeana vocabularies. It performs validation, normalisation, transformation, mapping of metadata and may need to interact with external resources.

This component needs access to all internal resources.

It needs to be able to read and import various kinds of external resources (schemas, ontologies, Wikipedia, provided these have a coherent structure and syntax (preferably expressed in XML) and with associated descriptive information (schema, DTD or similar). The architecture needs to be designed in such a way that also richer formats can be handled.

5.4.2 Constraints and assumptions

It is expected that the aggregator will provide content providers with a client implementation facilitating data collection in a form compliant to the Europeana needs. The metadata being harvested will be compatible with Europeana objectives and technical requirements.

The mapping of content provider metadata into the content aggregator metadata format will have to be carefully assessed.

5.5 Managing Europeana Objects

5.5.1 Description

Responsibility: implementing the Europeana surrogate model.

In the assumption that Europeana will manage a large set of information for each object, the object manager can conceptually be supported by the following components:

- **Content Manager**: responsible for analysing and pre-processing the digital content, and mapping the incoming content to Europeana data structures, and preparations for the Surrogates Manager. Because of the great heterogeneity that is to be expected on content, the content manager must be able to understand any content model, ranging from MPEG21[70] to PDF[71], to all sorts of proprietary models. The solution may be based on the content model underlying the Java Content Repository API, which is becoming a de facto standard for content repositories. The BRICKS Content Manager can then be considered as a candidate implementation.
- **Metadata Manager**: responsible for managing metadata at all levels: ontologies, schemas and records. Semantic Web technologies should be used for interoperability and generality. Several implementations exist, and in particular the Jena implementation seems a very viable option. The BRICKS Metadata Manager can then be considered as a candidate implementation.
- **Surrogates Manager**: responsible for supporting the creation of surrogates, including the Surrogate Root Component, as described in chapter 4.1.2, as well as all functionalities that each surrogate makes available.
- From the surrogates, various variants of presentations can be generated, including 'landing pages' for display and for consumption by search engines, either by presenting these pages on-the-fly or by pre-compiling them.
- **Rendering Manager**: responsible for creating output pages, depending also on the device where the pages are to be delivered. It may also provide links to additional information, such as advertising (e.g. events related to user profile or to results), Google Maps and Wikipedia when this is relevant to the object selected. It will get XML streams from other components and produce output depending on the channel (e.g. PC mobile device). This includes preparation of content for use on accessibility devices, such as screen readers, Braille displays, etc.

---

[70] MPEG21: http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm 24

[71] PDF – Portable Document Format: http://en.wikipedia.org/wiki/Portable_Document_Format

- **User Content Manager**: responsible for managing the user content within Europeana, either for single-user or collaborative spaces. This includes the management of virtual collections, baskets with selected items, folksonomies and the like.

5.5.2 Constraints and assumptions

It is assumed that the information pertaining to the Europeana objects and surrogates will be organised and stored using RDF-based technology. Considering the enormous volume of RDF triples to be stored in the Europeana production environment much care should be given to scalability and performance issues: testing and optimising storage and indexing components is crucial in this respect.

Fallback solutions must be devised to eventually cope with unsatisfactory performance of the system: one of the fallback strategems thus could be to partition data in such a way as to create chunks of data that can then be processed in parallel or even serially by search operations: even though less satisfactory in functional terms this may be necessary to guarantee acceptable system performance.

5.6 Search, Exploration and Discovery

5.6.1 Description

Responsibility: supporting the discovery of information.

Assuming users would find useful both a simple (Google-like) and an advanced (database-like) search, the discovery manager will be supported by the following components:

- **Simple search Manager**: this is the component responsible for evaluating simple queries, possibly after expanding them for crossing language or other semantic boundaries. For personalising the object ranking on the basis of user preferences, the Personalisation Manager may be involved.
- **Advanced search Manager**: this is the component responsible for scoring advanced queries, involving attribute values, comparison operators and Boolean connectives. An important issue here is the query language that will be supported. A possible choice, consistent with the above suggestion of using Semantic Web languages for metadata, is to use an RDF query language, e.g. SPARQL which is supported by Jena. The BRICKS Query Mediator can be used as a basis for implementing both simple and advance search.
- **Search gateway**: responsible for reaching on-the-fly (i.e. at query evaluation time) sources which are not harvestable.
- **Language Manager**: this component will handle all functions related to Language, such as query and result translation, management of user-defined dictionaries and the like. It will link to external resources, such as dictionaries, thesauri and translators. For a further description of multilingual issue see chapter 6.
- **Result Manager**: implements all functionalities offered on the result of a discovery. It must allow the separate display of results of queries evaluated against external resources, such as Wikipedia.
- **Navigation Manager**: manages user interaction sessions, keeping track of the history of visited pages, possibly including external pages. Exploring the Europeana information space by content similarity can also be supported.

A detailed presentation of the use cases that each of these components must support is now given, based on the requirements collected in EDLnet.

5.6.2 Simple Search Manager

Figure 5presents the use cases of the Simple Search Manager. For simplicity, the initiator of the main use cases has been named simply User, but it should be clear that we understand rather an application to perform the role of the initiator. The same convention has been applied for all use cases presented in this Section. A Textual illustration of the main use cases follows.

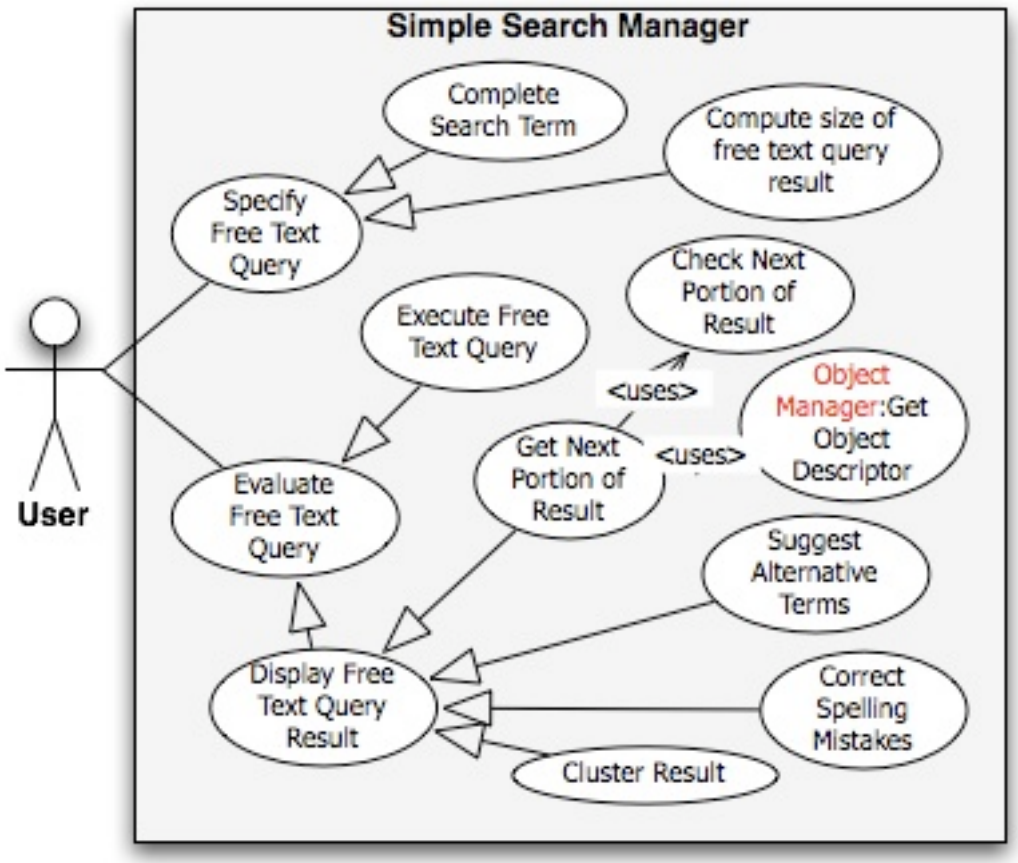

Figure 5: Simple Search Manager Use Cases

A Simple Search interaction is based on a free text query, and consists of two main parts, captured in as many use cases:

1. **Specify Free Text Query**; this use case is in fact an abstraction consisting of two concrete use cases:
    a. **Complete Search term**: while the using is typing each term in a query, the system automatically completes the term thus helping the user while avoiding as much as possible spelling mistakes;
    b. **Compute size of free text query result**, directly indicating at query specification time how many candidate objects a term retrieves.
2. **Evaluate Free Text Query**, a complex use case, using the following use cases:
    a. **Execute Free Text Query,** which is used in order to obtain the scoring of the query; typically, the evaluation function returns a handle for later accessing the result that has been generated as an effect of this use case;

this style of interaction is used in databases to alleviate the task of the caller, which would otherwise be a very sophisticated component, able to handle arbitrarily large results;

b. **Display Free Text Query**; also this is an abstract use case, that has a fundamental split between the case in which the result is non-empty, and the case in which the result is empty. In the former case, the result is displayed in successive stages, getting it at each stage via **Get Next Portion of Result**, which uses the **Check Next Portion of Result** to first test whether there are more results to display; in this context, **Get Object Descriptor** is used to obtain from The Object Manager the descriptor of each object to be displayed. Optionally, **Cluster Result** can be used to organize the objects to be displayed according to predetermined categories. If no meaningful result has been retrieved by query evaluation, the user is supported in creating a different query, in two different ways: by checking whether the current query terms are spelled correctly and, if not, suggesting corrections (**Correct Spelling Mistakes**); or, by proposing an alternative query which has a larger answer than the current one (**Suggest Alternative Terms**).

5.6.3 Advanced Search Manager

The functions of the Advanced Search Manager (see Figure 6) are similar to those of the Simple Search Manager. Also in this case, there are two main use cases, an abstract one for query specification (**Specify Advanced Query**) and an abstract one for query evaluation (**Evaluate Advanced Query**).

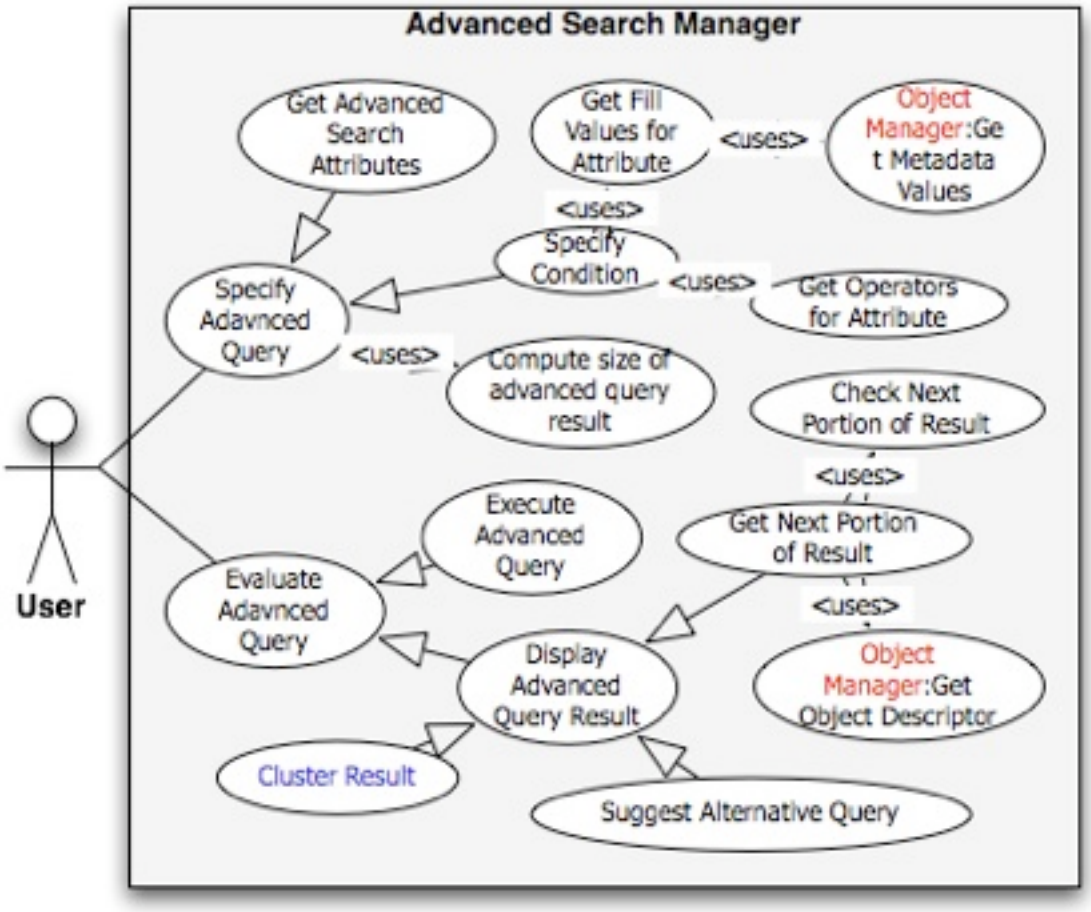

Figure 6: Advanced Search Manager Use Cases

In order to support advanced query specification, the **Get Advanced Search Attributes** use case provides the list of queryable attributes, while **Specify Condition** is an aid for expressing a condition on a given attribute: the list of possible operators (**Get Operators for Attribute**) or the corresponding values (**Get Fill Values for Attribute**) are provided.

For query evaluation, we have the **Execute Advanced Query** and **Display Advanced Query Results** use cases. The latter is further specialized according to the same criterion for Simple Search: (non-empty) results are obtained in two stages by using the same use cases, while the support for coping with empty results is different and only consists of one use case, **Suggest Alternative Query**, with the obvious meaning.

In addition, the Advanced Search Manager offers the functionality to support browsing along three dimensions: the Space, Time and Who & What dimensions (Figure 7).

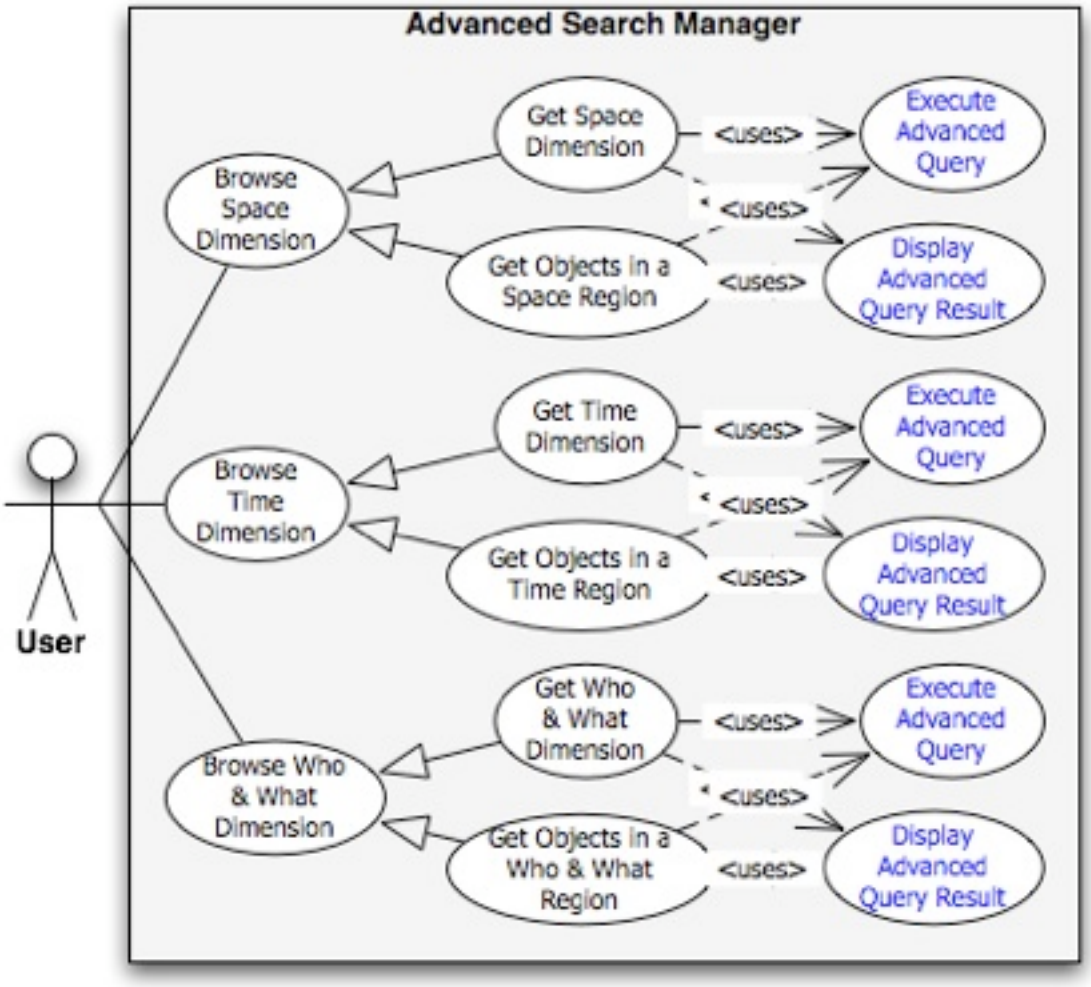

Figure 7: Use Cases for Browsing

For each dimension, there are two use cases:

- one for obtaining the information required for the initial display (**Get Space/Time/Who & What Dimension**); in order to obtain this information, an advanced search is executed, based on a query that denotes the required objects;
- one for retrieving the object(s) requested by the user via a selection on a region (**Get Objects Space/Time/Who & What Region**), translated as an advanced query.

All use cases re-use the already defined ones for executing an advanced search and displaying the query results. In fact, browsing is just a different way of specifying an advanced query and consuming its results.

5.6.4 Result Manager

The Result Manager offers three main use cases (Figure 8):

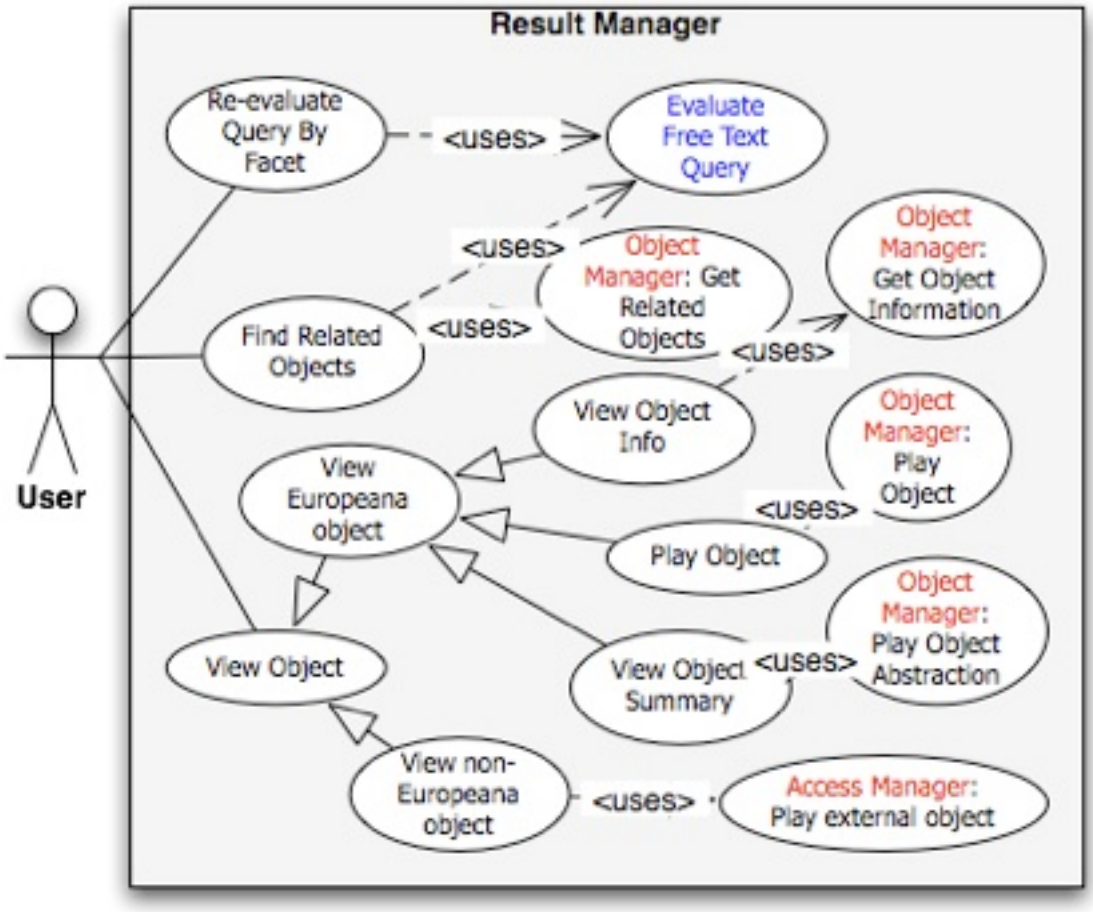

Figure 8: Result Manager Use Cases

- **Re-evaluate Query By Facet**, allows users who are consuming the results of the current query to include in the query a selected facet and execute the resulting query, thus oriented towards a specific direction; for the evaluation of the new query the **Evaluate Free Text Query** use case is employed.
- **Find Related Objects,** allows users who have selected one object in a result set, to obtain related objects; these objects may be gathered in one of two different ways: (a) by expressing the request as a free text query and execute the resulting query; in this case, the **Evaluate Free Text Query** use case is involved; or (b) by retrieving the requested objects from the digital library; in this case, the Object Manager is involved.
- **View Object,** allows users to access a selected object. This is an abstract use case, further structured in two abstractions, corresponding to the cases in which the selected object is or not a Europeana object. **View non-Europeana Object** invokes the Access Manager to play the selected external object. **View Europeana Object** is further divided into three sub-cases:
    - **View Object Info,** providing (a convenient subset of) the Europeana surrogate of the selected object;
    - **View Object Summary,** playing an abstraction (thumbnail, trailer and the like) of the selected object;
    - **Play Object,** offering the fruition of the object.

In addition, the Result Manager offers the functionality for saving a just executed query or a browsing experience (see Figure 9**Fehler! Verweisquelle konnte nicht gefunden werden.**).

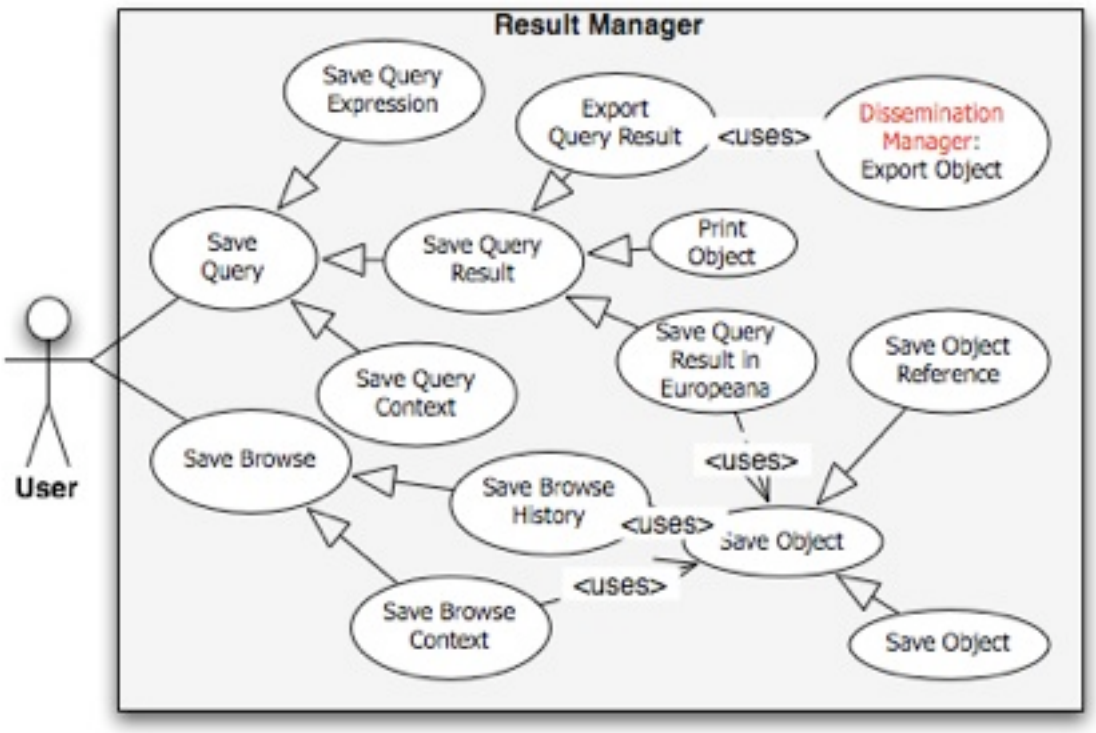

Figure 9: Result Manager (Advanced)

Save Query is categorized in 3 use cases, corresponding to the cases in which the query expression (**Save Query Expression**) or context (**Save Query Context**) or result (**Save Query Result**) is saved. For the first two cases, the save can be done either within the user space in MyEuropeana, or within the space of a community where the user belongs. In contrast, the result of a query can be also saved outside Europeana, that means exported. Consequently, **Save Query Result** is specialized in:

- **Export Query Result**, which invokes the Dissemination Manager to allow the download of a Europeana object;
- **Print Object**, to obtain a hard copy of textual objects;
- **Save Query Result in Europeana**, which uses **Save Object**.

In turn, Save Object is specialized in **Save Object Reference** and **Save Object** to store a reference to the selected object or the content itself respectively, either in the user private space within MyEuropeana or in a community space.

1

5.6.5 Navigation Manager

As already pointed out, the Navigation Manager (see Figure 10) supports the exploration of result sets, by allowing a user to move to the previous/next object in a set or in a carousel, to the next related object, or to the next component in the

structure of the current object. Each of these four use cases, uses **View Object** (see above) to actually see the so selected object.

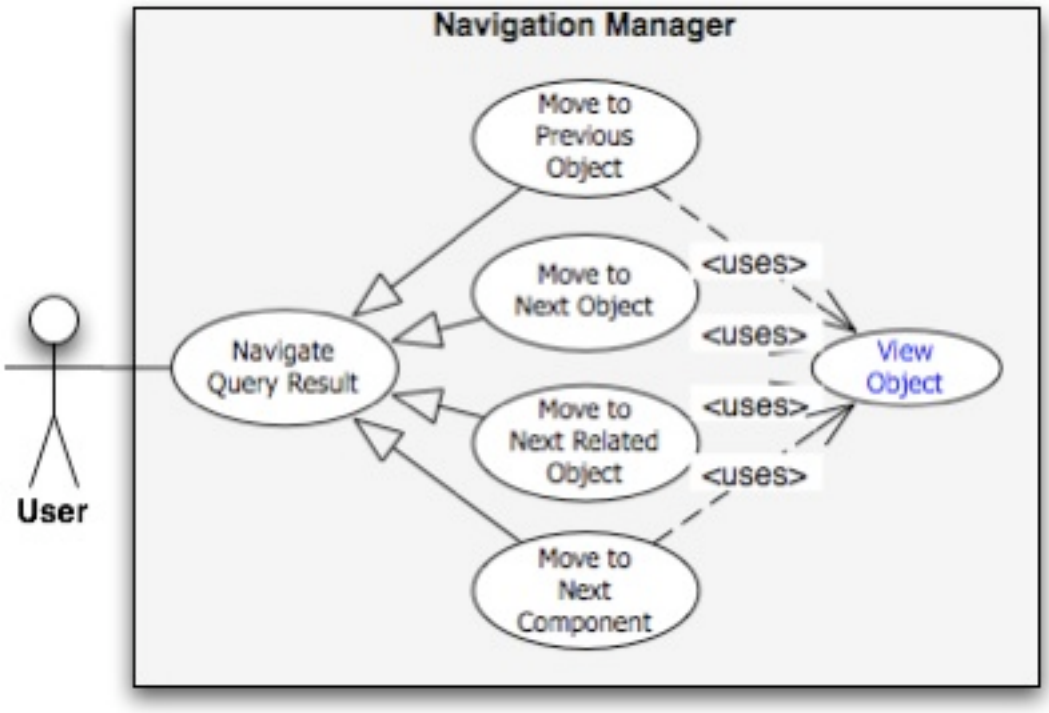

Figure 10: Navigation Manager

5.6.6 External Search Manager (Search gateway)

The External Search Manager has only two use cases, corresponding to the two types of queries that can be expressed by the user. In each case, the task of evaluating the query and translating back its result is carried out by the External Service Manager.

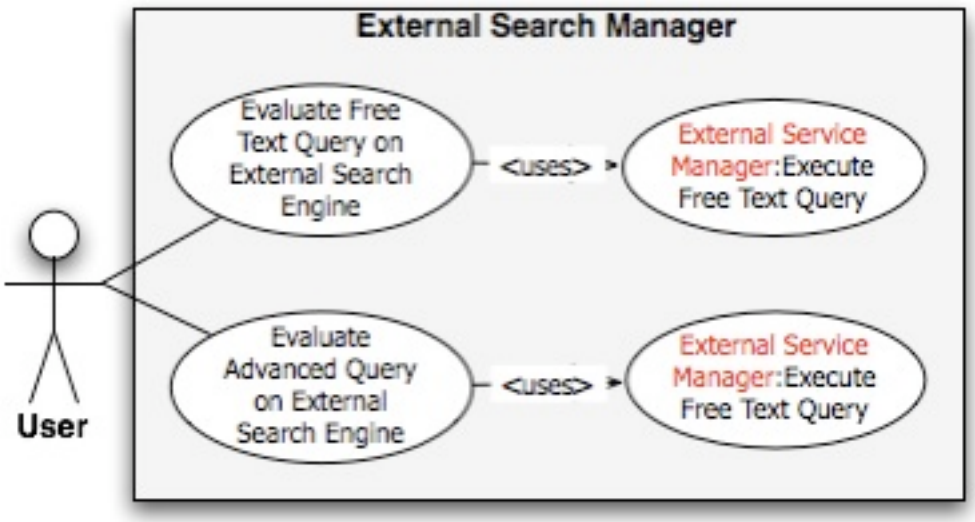

Figure 11: External Search Manager Use Cases

5.6.7 Community Manager

The uses cases of the Community Manager (see Figure 12**Fehler! Verweisquelle konnte nicht gefunden werden.**) have two different actors, the Registered User and the Community Administrator.

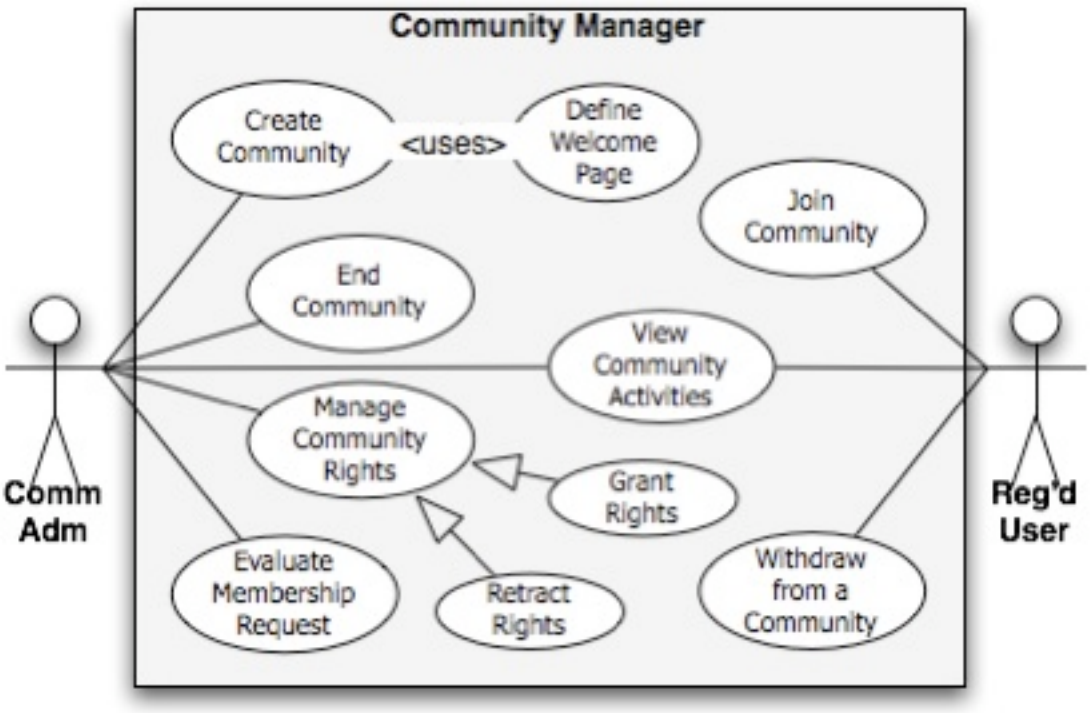

Figure 12: Community Manager Use Cases

A registered user can join or withdraw from a community and view the activities that have been so far carried out within their community.

A community administrator can create or end a community, manage the rights for administering a community (**Grant Rights** or **Retract Rights**), and is the one who evaluates the requests to join a community. An administrator can also view the community activities.

5.6.8 User Content Manager (MyEuropeana)

The User Content Manager offers three main types of functionality:

- to manage access to the user space in MyEuropeana; this result in three use cases:
  - **Register to MyEuropeana**, to create new users;
  - **Login / Logout MyEuropeana**;
  - **View / Edit user profile**, resolved to the Profile Manager.
- to manage the subscription to the Europeana services; theis result in three use cases:
  - **Subscribe / Unsubscribe to Service**;
  - **View User Subscriptions**;

- **View Available Services**.

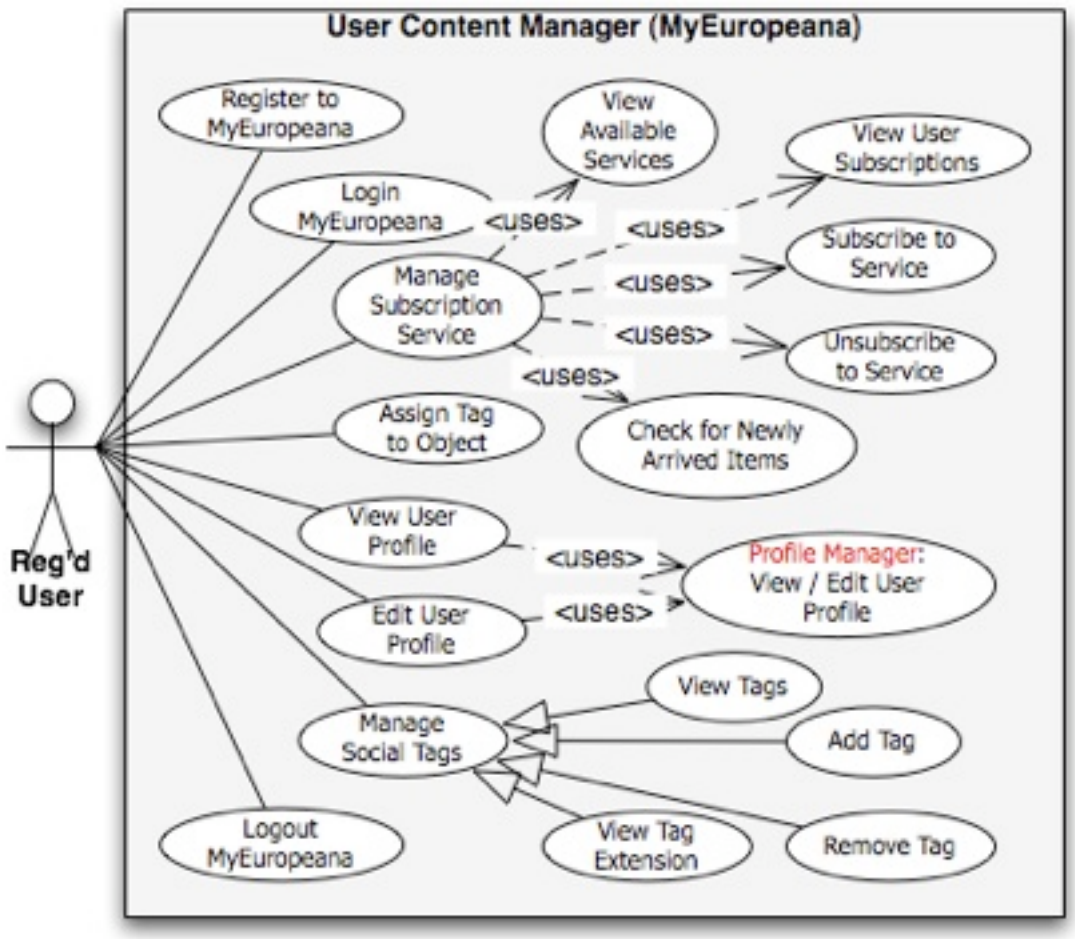

Figure 13: User Content Manager Use Cases

- to assign and manage social tags; these are the use cases for:
  - Add / Remove a Tag;
  - View Tags;
  - View Tag Extension, that is the objects which are assigned a given tag

5.7 Managing Users

5.7.1 Description

Responsibility: managing user profiles and permissions, including personalisation. Its main sub-components are:

- **Authentication Manager**: responsible for authenticating users, implementing single sign-on and integrating with the European infrastructure under development by Terena.
- **Authorisation Manager**: responsible for managing which users can perform which operations on which objects.
- **Profile Manager**: responsible for managing user models, including preferences

- **Personalisation Manager**: enforcing preferences, e.g. on the results of search operations, or on browsing
- **Community Manager**: supporting communities of users created for purposes of collaborative work.

5.7.2 Constraints and assumptions

Authentication methods chosen must be SAML-compliant in order to enable trust based interaction between Europeana, its partners and external instances. More generally, the authentication and authorisation framework should be designed and implemented in such a way as to reuse as much as possible the results of Géant 2 and more specifically comply to the AAI architecture[72] proposed there as well as take into account the best practices provided in the Géant AAI cookbook[73].

5.8 Managing Access

5.8.1 Description

Responsibility: responsible for accessing content on external sources. It offers functionality for permissions and trust management. Its main sub-components are:

- **Object Identity Manager**: supporting the creation of Europeana identifiers, the mapping to external identifiers, and the involved resolution process.
- **Trust Manager**: manages the trust between Europeana and the external sources with which Europeana must interact for various purposes (SAML standard based models such as Shibboleth may be adopted for this functionality)
- This component needs a list of external resources (e.g. providers) and external services and collections
- **Digital Rights Manager**
- **Remote Access Manager**: managing access to digital objects. Objects sitting behind URLs do not require much work. If reaching the object requires instead a complex interaction, it is the responsibility of the Remote Access Manager to support this interaction.
- **External Service Manager**: managing the interaction with external service providers.

5.8.2 Constraints and assumptions

Here again, the findings of Géant 2 should be the guideline for Europeana detailed specification and later implementation.

---

[72] http://www.geant2.net/upload/pdf/GN2-07-024-DJ5-2-2-2-GEANT2_AAI_Architecture_And_Design.pdf

[73] http://www.geant2.net/upload/pdf/GN2-07-023v4-DJ5-2-3_2_Best_Practice_Guide-AAI_Cookbook-Second_Edition.pdf

5.9 System Administration

5.9.1 Description

The system administration component will be responsible for managing the operational environment to run continuously. To satisfy the need to give access to Europeana 24 hours a day every day of the year, it will be necessary to separate the production and pre-processing environments.

This component needs to manage administrative information (service registry, user administration and history) and should be able to import external data (e.g. statistics on provider sites) and to export data to reporting systems. Remote and secure access for system administrators should be supported.

5.9.2 Constraints and assumptions

6 Multilingual issues

In a first theoretical approximation, four levels of potential implementation regarding language interoperability can be identified.

6.1 Interface

The most elementary level of language interoperability (and probably the one that should be achieved as part of the first prototype) is an interface localised in the major European languages. A basic software design requirement for meeting this goal is a systematic and sufficient separation of language token administration from development of other code elements. This so-called 'language skinning' should be scalable with regards to addition of new languages and have a maintenance mechanism in place that deals with the translation of new terms. For example, when a new function becomes available a call for translation is put out to the responsible translators. The European Library currently has interface translations available in 21 languages. This provides a good starting point for Europeana.

6.2 Browsing

Another way of achieving a reasonably simple language interoperability (or access over languages) would be to enable browsing via a common multilingual ontology mapping onto versions for each language; for the first prototype this could be implemented at a fairly high level.

Via SKOSification and alignment of multilingual thesauri and ontologies, a multilingual backbone could be created for Europeana. Providing the user with native language support at the interface, browse and result level. At the result level, metadata with controlled vocabulary could be presented in the language of the interface. Merging and aligning of name authority files is also important to enhance cross-language interoperability. The user should be able to find and browse all works by a given author in the orthographic representation he is familiar with.

With regards to multilingual browsing Europeana can build on established multilingual initiatives, like MACS[74], CRISSCROSS[75], MSAC[76], Michael+[77] and VIAF[78].

---

[74] MACS – Multilingual Access to Subjects: https://macs.vub.ac.be/pub/

[75] CrissCross: http://www.d-nb.de/eng/wir/projekte/crisscross.htm

6.3 Search on a monolingual baseline

By multilingual search in Europeana, ideally we intend three levels of access: 1. efficient monolingual search in all languages supported by Europeana; 2. simple cross-language access (querying in one language (L1) against a target collection in a second language (L2); 3. multilingual access (querying on L1 against target collections in Ln, where n=all languages supported by Europeana). The interface of the maquette should demonstrate all three functionalities.

(1.) and a preliminary version of (2.) will be implemented in the first public prototype.

6.3.1 Monolingual Search for Multiple Languages

For each language supported by Europeana, language-specific processing and indexing tools must be available. These include tokenizers (essential), stopword lists (essential), stemmers or morphological analysers (essential), decompounders (optional), phrase recognizers (optional), named entity recognizers (optional).

6.3.2 Simple Cross-language Search

In order to match queries to documents, state-of-the-art cross-language retrieval systems use both query and document translation methodologies. In the former, the query is translated into the language of the document; in the latter the document collection is translated (using a Machine Translation (MT) system into the language of the query). There are pros and cons to both approaches.

For simple cross-language search, query translation is the method most commonly adopted and we recommend it here. Both MT and bilingual dictionary based techniques can be used (at times a combination of the 2 techniques is employed). MT techniques are preferable if the requirement is an automatic translation (transparent to the user); dictionary based methods should be used for interactive systems, where the user is given the opportunity to select (or modify) the preferred translation(s) from the set proposed by the dictionary. This is the method proposed for Europeana.

In order to implement this method, machine-readable bilingual dictionaries for all language pairs supported by Europeana are needed. Many open-source dictionaries are now available and can be downloaded from Internet; however, considerable effort is needed to build a set of dictionaries with sufficiently comprehensive coverage. For language pairs where no bilingual dictionary is available, a pivot language can be used (L1 -> Lpivot -> L2); this is not advisable for the preliminary implementation.

In order to improve results, query and/or document expansion methods can be used.

6.3.3 Multilingual Search

It should be clear from the above that the implementation of cross-language or multilingual search in Europeana requires considerable efforts and resources. We recommend that the maquette should provide clear examples of the full functionality; however, it cannot be expected to implement more than a reduced cross-language search by November 2008.

---

[76] MSAC – Multilingual Subject Access to Catalogues of National Libraries: http://info.jib.cz/dokumenty/msac_metod.pdf

[77] Michael+: http://www.d-nb.de/eng/wir/projekte/michael.htm

[78] VIAF – Virtual International Authority File: http://viaf.org/

Many of the issues to be faced by Europeana are already being addressed from a different perspective by the MultiMatch project[79]. We recommend that collaborations and information exchanges are established between these two projects. The Cross-Language Evaluation Forum[80] intends to experiment with cross-language search on library catalogue data in CLEF 2008 using data provided by TEL. The results of these experiments should provide useful input for Europeana.

6.4 Result translation

In a rather ambitious and resource consuming and 'expensive' setting, query translation approaches should be complemented with result translation. This can be achieved at two levels: at the metadata – or even more demanding – at the digital objects level. Given the limits of the Europeana surrogate model as specified hereafter the only targets within reach even in a medium term approach probably are metadata and abstractions (or more generally any other of the surrogate components in general). Suitable templates can be produced for each language and domain type onto which translated metadata can be matched appropriately in order to provide the user with a descriptive snippet in the query language for each document (digital object) retrieved. The propagation of result translation methods at the level of federated actual objects probably will remain utopian for quite some time from here and may even not be desirable at all, because the information behind the translated terms will be in this foreign language, too. A translation of these texts using machine translation methods (e.g. Babel Fish[81]) produces sometimes entertaining but in most cases but not reasonable or useful results, at least if a vocabulary above a thousand words is used.

.

7 References

All web links are given when first mentioning a given item. The following bibliographic references have been used:

[Amato_et_al2003] Giuseppe Amato, Fausto Rabitti, Pasquale Savino, Pavel Zezula: Region proximity in metric spaces and its use for approximate similarity search. ACM Trans. Inf. Syst. 21(2): 192-227 (2003)

[Amato_et_al2007] Giuseppe Amato, Carlo Meghini, "Combining Features for Image Retrieval by Concept Lattice Querying and Navigation" , VMDL 2007, International Workshop on Visual and Multimedia Digital Libraries, In conjunction to ICIAP 2007, Modena, 10-14 September 2007

[Ciaccia_et_al1997] Paolo Ciaccia, Marco Patella, Pavel Zezula: M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. VLDB 1997: 426-435

[Falchi_et_al2007] Fabrizio Falchi, Claudio Gennaro, Pavel Zezula: Nearest neighbor search in metric spaces through Content-Addressable Networks. Inf. Process. Manage. 43(3): 665-683 (2007)

[Guttmann1984] Antonin Guttman: R-Trees: A Dynamic Index Structure for Spatial Searching. SIGMOD Conference 1984: 47-57

---

79 MultiMatch project: http://www.multimatch.eu/

80 CLEF – Cross-Language Evaluation Forum: www.clef-campaign.org

81 Babel Fish: http://babelfish.altavista.com/

[LeSaux_et_al2004] Bertrand Le Saux, Giuseppe Amato, "Image Classifier for scene analysis" ,ICCVG 04, International Conference on Computer Vision and Graphics, Warsaw, Poland September 22-24, 2004

[Samet2006] Hanan Samet, Foundations of Multidimensional and Metric Data Structures. Amsterdam: Elsevier, 2006

[Zezula_et_al1998] Pavel Zezula, Pasquale Savino, Giuseppe Amato, Fausto Rabitti: Approximate Similarity Retrieval with M-Trees. VLDB J. 7(4): 275-293 (1998)

[Zezula_et_al2006] Pavel Zezula, Giuseppe Amato, Vlastislav Dohnal, Michal Batko, "Similarity Search - The Metric Space Approach", Heidelberg: Springer, 2006. Series: Advances in Database Systems, Vol. 32

8 Index

9 Acknowledgements

This document is the result of discussions that have taken place in meetings of the WP2 working groups between January and May2008 and on the EDLnet workgroup mailing list. The members of these Working Groups were:

Leif Andresen (Danish Library Agency, Denmark), Emmanuelle Bermes (Bibliothèque nationale de France), Marco Berni (Istituto e Museo di Storia della Scienza, Italy), Birte Christensen-Dalsgaard (Danish State Library Aarhus, Denmark), Santiago Chumbe (Heriot-Watt University, UK), Robina Clayphan (British Library, UK), Milena Dobreva (Institute of Mathematics, Bulgaria), Jean-Pierre Evain (EBU, Switzerland), Licia Florio (Terena, Netherlands), Michael Fuhr (Ethnologic Museum Berlin, Germany – project DISMARC), Juha Hakala (National Library of Finland), Antoine Isaac (Vrije Universiteit, Netherlands), Ullrich Kampffmeyer (Project Consult, Germany – DLM Forum), Stefanos Kollias (National Technical University of Athens, Greece), Marco de Niet (Digitaal Erfgoed Nederland, Netherlands), Patrick Peiffer (National Library of Luxembourg), Carol Peters (IEI-CNR, Italy), Vivien Petras (GESIS, Germany), Stéphane Pillorget (Bibliothèque nationale de France), Neil Thomson (Natural History Museum, UK), Vassilis Tzouvaras (National Technical University of Athens, Greece – project Video Archive), Theo van Veen (National Library of the Netherlands)